5K BASIC
(SOFTWARE #2)

Processor Technology

6200 Hollis Street
Emeryville, CA 94608
Phone: (415) 652-8080

```
0000                *
0000                *
0000                *
0000                *               PROCESSOR TECHNOLOGY  BASIC-5
0000                *          ==--=====================================--==
0000                *
0000                *
0000                *  DEVELOPED IN CONJUNCTION WITH:
0000                *
0000                *          APPLIED COMPUTER TECHNOLOGY
0000                *          1038 MERCED STREET
0000                *          BERKELEY, CALIF    94707
0000                *         ******************************
0000                *
0000                *
0000                *     ASSEMBLED USING THE ALS-8 PROGRAM DEVELOPMENT SYSTEM
0000                *
0000                *
0000                *              ***   COPYRIGHT 1976   ***
0000                *
0000                *                 PROCESSOR TECHNOLOGY CORP.
0000                *                 6200 HOLLIS STREET
0000                *                 EMERYVILLE, CALIF.
0000                *                 94608     (415) 652-"8080"
0000                *
0000                *           <<<<<<<<- ALL RIGHTS RESERVED ->>>>>>>>
0000                *
0000                *
0000                *      SYSTEM GLOBAL EQUATES
0000                *
0000                FPSIZ    EQU     5
0000                LINLEN   EQU     73          # CHRS IN LEGAL INPUT LINE
0000                FP123    EQU     FPSIZ-2
0000                FPNIB    EQU     FP123*2
0000                DIGIT    EQU     FPNIB/2
0000                CR       EQU     15Q         CARRIAGE RETURN
0000                NULL     EQU     0           NULL CHARACTER VALUE
0000                LF       EQU     12Q         LINE FEED
0000                ESC      EQU     3Q          ESCAPE CHAR
0000                EOF      EQU     1           END OF FILE
0000                BELL     EQU     7           BELL CHARACTER
0000                STESIZ   EQU     2+FPSIZ     SYMBOL TABLE ELEMENT SIZE
0000                OPBASE   EQU     '('
0000                FTYPE    EQU     1           CONTROL STACK FOR ENTRY TYPE
0000                FORSZ    EQU     FPSIZ*2+2+2+1  'FOR' CONTROL STACK ENTRY SIZE
0000                GTYPE    EQU     2           CONTROL STACK GOSUB ENTRY TYPE
0000                ETYPE    EQU     0           CONTROL STACK UNDERFLOW TYPE
0000                UMINU    EQU     61Q         UNARY MINUS
0000                *
0000                *       STARTUP BASIC SYSTEM
0000                *
0000 31 90 17       START    LXI     SP,CMNDSP
0003 3E 03                   MVI     A,3
0005 32 92 17                STA     NULLCT      INITIALIZE NULL COUNT
```

```
0008 32 45 19                 STA     SPEED
000B CD 68 0F                 CALL    INITS
000E 11 6A 07                 LXI     D,FAS       FIRST ADDRESS MESSAGE
0011 21 53 19                 LXI     H,MEMTO+1   ONE LESS THAN LOWER BOUND
0014 CD 3B 10                 CALL    GINT        GET INTEGER GREATER THAN MEMTOP+1
0017 22 50 19                 SHLD    BOFA        START OF USER ASSIGNED MEMORY
001A 11 76 07                 LXI     D,LAS
001D CD 3B 10                 CALL    GINT        HL HAS 1 LESS THAN LOWER BOUND AGAIN
0020 22 52 19                 SHLD    MEMTOP      END OF ASSIGNED MEMORY POINTER
0023 21 43 07         STO     LXI     H,PLS       "PROGRAM LOADED?" MESSAGE
0026 CD 23 0D                 CALL    PRNT
0029 CD 55 0E                 CALL    INLINE
002C 3A D3 18                 LDA     IBUF
002F                  *
002F                  *     OPTIONAL ENTRY POINT FOR TAPE OR DISK ROUTINES
002F                  *
002F                  *  ALLOWS DIRECT PROGRAM INPUT FROM HIGH SPEED DEVICES
002F                  *  SEE OPERATING INSTRUCTIONS FOR PROPER IMPLEMENTATION.
002F                  *
002F FE 4E            STAR1   CPI     'N'
0031 CA 51 00                 JZ      ST1         IF NO PROGRAM CLEAR AND INITIALIZE
0034 FE 59                    CPI     'Y'
0036 C2 23 00                 JNZ     STO
0039 2A 50 19                 LHLD    BOFA
003C 7E               ST2     MOV     A,M         FIND END OF PROGRAM
003D FE 01                    CPI     EOF
003F CA 48 00                 JZ      ST3
0042 CD C1 01                 CALL    ADR
0045 C3 3C 00                 JMP     ST2
0048 22 4E 19         ST3     SHLD    EOFA
004B CD DF 01                 CALL    CCLEAR
004E C3 54 00                 JMP     ST4
0051 CD D7 01         ST1     CALL    CSCR
0054 3E 0C            ST4     MVI     A,2*FPNIB
0056 32 3A 19                 STA     INFES
0059                  * INITIALIZE RANDOM NUMBER
0059 11 CF 18                 LXI     D,FRAND
005C 21 B1 07                 LXI     H,RANDS
005F CD 9C 0C                 CALL    VCOPY       FRAND=RANDOM NUMBER SEED
0062                  *
0062                  *         COMMAND PROCESSOR
0062                  *
0062 CD 19 10         CMND1   CALL    CRLF2
0065 21 3D 07                 LXI     H,RDYS      PRINT READY MESSAGE
0068 CD 23 0D                 CALL    PRNT
006B 3E 01            CMNDR   MVI     A,1         SET DIRECT INPUT FLAG
006D 32 94 17                 STA     DIRF
0070 31 90 17                 LXI     SP,CMNDSP
0073 CD 1C 10                 CALL    CRLF
0076 CD 55 0E         CMND2   CALL    INLINE      GET INPUT LINE FROM OPERATOR
0079 CD 9F 06                 CALL    PP          PRE-PROCESSOR IT
007C DA 88 00                 JC      CMND3
007F CD 10 01                 CALL    LINE        LINE NUMBER..GO EDIT
0082 CD DF 01                 CALL    CCLEAR
0085 C3 76 00                 JMP     CMND2
0088 CD 8E 00         CMND3   CALL    CMND4
```

```
008B C3 6B 00                  JMP     CMNDR
008E 21 D3 18         CMND4    LXI     H,IBUF       POINT TO COMMAND OR STATEMENT
0091 22 95 17                  SHLD    TXA
0094 CD 02 0D                  CALL    GC
0097 E6 A0                     ANI     240Q
0099 FE A0                     CPI     240Q         CHECK FOR COMMAND
009B 11 D2 07                  LXI     D,CMNDD
009E CA 8E 02                  JZ      ISTA1        PROCESS COMMAND
00A1 CD 7F 02                  CALL    ISTAT        PROCESS STATEMENT (IF ALLOWED)
00A4 CD 0A 0D                  CALL    GCI
00A7 FE 0D                     CPI     CR
00A9 C8                        RZ
00AA 01 53 42         E1       LXI     B,'BS'
00AD C3 C5 00                  JMP     ERROR
00B0                  * ERROR MESSAGE PRINT OUT
00B0 01 41 42         E3       LXI     B,'BA'
00B3 C3 C5 00                  JMP     ERROR
00B6 01 53 43         E4       LXI     B,'CS'
00B9 C3 C5 00                  JMP     ERROR
00BC 01 42 4F         E5       LXI     B,'OB'
00BF C3 C5 00                  JMP     ERROR
00C2 01 4D 44         E6       LXI     B,'DM'
00C5                  *
00C5 C5               ERROR    PUSH    B
00C6 CD 1C 10                  CALL    CRLF
00C9 C1                        POP     B
00CA CD 95 0E                  CALL    CHOUT
00CD 41                        MOV     B,C
00CE CD 95 0E                  CALL    CHOUT
00D1 21 54 07                  LXI     H,ERS
00D4 CD 23 0D         ERM1     CALL    PRNT
00D7 3A 94 17                  LDA     DIRF
00DA B7                        ORA     A
00DB C2 62 00                  JNZ     CMND1
00DE 21 5B 07                  LXI     H,INS
00E1 CD 23 0D                  CALL    PRNT
00E4                  * FIND LINE NUMBER
00E4 2A 50 19                  LHLD    BOFA
00E7 44               ERM2     MOV     B,H
00E8 4D                        MOV     C,L
00E9 5E                        MOV     E,M
00EA 16 00                     MVI     D,0
00EC 19                        DAD     D
00ED EB                        XCHG
00EE 21 95 17                  LXI     H,TXA
00F1 CD 35 0D                  CALL    DCMP
00F4 EB                        XCHG
00F5 DA E7 00                  JC      ERM2
00F8 03                        INX     B
00F9 0A                        LDAX    B
00FA 6F                        MOV     L,A
00FB 03                        INX     B
00FC 0A                        LDAX    B
00FD 67                        MOV     H,A
00FE 11 D3 18                  LXI     D,IBUF       USE IBUF TO ACCUMULATE THE LINE NUMBER STRING
0101 CD FA 0D                  CALL    CNS
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
0104 3E 0D             MVI     A,CR
0106 12                STAX    D
0107 21 D3 18          LXI     H,IBUF
010A CD 1E 0D          CALL    PRNTCR
010D C3 62 00          JMP     CMND1
0110              *
0110              *     LINE EDITOR
0110              *
0110 2A 50 19     LINE LHLD    BOFA      CHECK FOR EMPTY FILE
0113 7E           FIN  MOV     A,M       CHECK IF APPENDING LINE AT END
0114 3D                DCR     A
0115 CA 2F 01          JZ      APP
0118 EB                XCHG
0119 13                INX     D
011A 2A D1 18          LHLD    IBLN      GET INPUT LINE NUMBER
011D EB                XCHG
011E CD 35 0D          CALL    DCMP      COMPARE WITH FILE LINE NUMBER
0121 2B                DCX     H
0122 DA 44 01          JC      INSR      LESS THAN
0125 CA 44 01          JZ      INSR      EQUAL
0128 7E                MOV     A,M       LENGTH OF LINE
0129 CD C1 01          CALL    ADR       JUMP FORWARD
012C C3 13 01          JMP     FIN
012F              * APPEND LINE AT END CASE
012F 3A D0 18     APP  LDA     IBCNT     DON'T APPEND NULL LINE
0132 FE 04             CPI     4
0134 C8                RZ
0135 CD C6 01          CALL    FULL      CHECK FOR ROOM IN FILE
0138 2A 4E 19          LHLD    EOFA      PLACE LINE IN FILE
013B CD 98 01          CALL    IMOV
013E 36 01             MVI     M,EOF
0140 22 4E 19          SHLD    EOFA
0143 C9                RET
0144              * INSERT LINE IN FILE CASE
0144 46           INSR MOV     B,M       OLD LINE COUNT
0145 22 3D 19          SHLD    INSA      INSERT LINE POINTER
0148 3A D0 18          LDA     IBCNT     NEW LINE COUNT
014B DA 5C 01          JC      LT        JMP IF NEW LINE # NOT = OLD LINE NUMBER
014E D6 04             SUI     4
0150 CA 55 01          JZ      LT1       TEST IF SHOULD DELETE NULL LINE
0153 C6 04             ADI     4
0155 90           LT1  SUB     B
0156 CA 8F 01          JZ      LIN1      LINE LENGTHS EQUAL
0159 DA 7B 01          JC      GT
015C              * EXPAND FILE FOR NEW OR LARGER LINE
015C 47           LT   MOV     B,A
015D 3A D0 18          LDA     IBCNT
0160 FE 04             CPI     4         DON'T INSERT NULL LINE
0162 C8                RZ
0163 78                MOV     A,B
0164 CD C6 01          CALL    FULL
0167 2A 3D 19          LHLD    INSA
016A CD B5 01          CALL    NMOV
016D 2A 4E 19          LHLD    EOFA
0170 EB                XCHG
0171 22 4E 19          SHLD    EOFA
```

```
0174 03              0223          INX       B
0175 CD AA 01        0224          CALL      RMOV
0178 C3 8F 01        0225          JMP       LIN1
017B                 0226 * CONTRACT FILE FOR SMALLER LINE
017B 2F              0227 GT       CMA
017C 3C              0228          INR       A
017D CD C1 01        0229          CALL      ADR
0180 CD B5 01        0230          CALL      NMOV
0183 EB              0231          XCHG
0184 2A 3D 19        0232          LHLD      INSA
0187 C4 9F 01        0233          CNZ       LMOV
018A 36 01           0234          MVI       M,EOF
018C 22 4E 19        0235          SHLD      EOFA
018F                 0236 * INSERT CURRENT LINE INTO FILE
018F 2A 3D 19        0237 LIN1     LHLD      INSA
0192 3A D0 18        0238          LDA       IBCNT
0195 FE 04           0239          CPI       4
0197 C8              0240          RZ
0198                 0241 * INSERT CURRENT LINE AT ADDR HL
0198 11 D0 18        0242 IMOV     LXI       D,IBCNT
019B 1A              0243          LDAX      D
019C 4F              0244          MOV       C,A
019D 06 00           0245          MVI       B,0
019F                 0246 * COPY BLOCK FROM BEGINNING
019F                 0247 * HL IS DESTIN ADDR, DE IS SOURCE ADDR, BC IS COUNT
019F 1A              0248 LMOV     LDAX      D
01A0 77              0249          MOV       M,A
01A1 13              0250          INX       D
01A2 23              0251          INX       H
01A3 0B              0252          DCX       B
01A4 78              0253          MOV       A,B
01A5 B1              0254          ORA       C
01A6 C2 9F 01        0255          JNZ       LMOV
01A9 C9              0256          RET
01AA                 0257 * COPY BLOCK STARTING AT END
01AA                 0258 * HL IS DESTIN ADDR,DE IS SOURCE ADDR, BC IS COUNT
01AA 1A              0259 RMOV     LDAX      D
01AB 77              0260          MOV       M,A
01AC 2B              0261          DCX       H
01AD 1B              0262          DCX       D
01AE 0B              0263          DCX       B
01AF 78              0264          MOV       A,B
01B0 B1              0265          ORA       C
01B1 C2 AA 01        0266          JNZ       RMOV
01B4 C9              0267          RET
01B5                 0268 *    COMPUTE FILE MOVE COUNT
01B5                 0269 * BC GETS (EOFA) - (HL), RET Z SET MEANS ZERO COUNT
01B5 3A 4E 19        0270 NMOV     LDA       EOFA
01B8 95              0271          SUB       L
01B9 4F              0272          MOV       C,A
01BA 3A 4F 19        0273          LDA       EOFA+1
01BD 9C              0274          SBB       H
01BE 47              0275          MOV       B,A
01BF B1              0276          ORA       C
01C0 C9              0277          RET
01C1                 0278 * ADD A TO HL
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
01C1 85         ADR     ADD     L
01C2 6F                 MOV     L,A
01C3 D0                 RNC
01C4 24                 INR     H
01C5 C9                 RET
01C6            * CHECK FOR FILE OVERFLOW, LEAVES NEW EOFA IN DE
01C6            * A HAS INCREASE IN SIZE
01C6 2A 4E 19   FULL    LHLD    EOFA
01C9 CD C1 01           CALL    ADR
01CC EB                 XCHG
01CD 21 52 19           LXI     H,MEMTOP
01D0 CD 35 0D           CALL    DCMP
01D3 D2 F4 0C           JNC     E8
01D6 C9                 RET
01D7            *
01D7            *         COMMANDS
01D7            *
01D7 2A 50 19   CSCR    LHLD    BOFA
01DA 36 01              MVI     M,EOF
01DC 22 4E 19           SHLD    EOFA
01DF            * "CLEAR"
01DF 2A 4E 19   CCLEAR  LHLD    EOFA        CLEAR FROM EOFA TO MEMTOP
01E2 23                 INX     H
01E3 22 1E 19           SHLD    MATA
01E6 EB                 XCHG
01E7 21 52 19           LXI     H,MEMTOP    END OF ASSIGNED MEMORY
01EA AF         CCLR1   XRA     A
01EB 12                 STAX    D
01EC CD 35 0D           CALL    DCMP
01EF 13                 INX     D
01F0 C2 EA 01           JNZ     CCLR1
01F3 2A 52 19           LHLD    MEMTOP
01F6 22 B3 18           SHLD    STA
01F9 21 FA 17           LXI     H,CSTKL+CSTKSZ-1
01FC 36 00              MVI     M,ETYPE
01FE 22 B5 18           SHLD    CSTKA
0201 21 B5 18           LXI     H,ASTKL+ASTKSZ+FPSIZ-1
0204 22 1C 19           SHLD    ASTKA
0207 C9                 RET
0208            *  "NULL"
0208 CD E0 0D   CNULL   CALL    INTGER
020B DA B0 00           JC      E3          NO ARGUMENT SUPPLIED
020E 7D                 MOV     A,L
020F 32 92 17           STA     NULLCT
0212 C3 62 00           JMP     CMND1
0215            * "LIST"
0215 CD 02 0D   CLIST   CALL    GC
0218 FE 0D              CPI     CR
021A 11 00 00           LXI     D,0
021D CA 23 02           JZ      CLO         JUMP IF NO ARG SUPPLIED
0220 CD E0 0D           CALL    INTGER      ERROR DEFAULT IS LIST
0223 2A 50 19   CLO     LHLD    BOFA
0226 7E         CL1     MOV     A,M
0227 3D                 DCR     A
0228 C8                 RZ
0229 23                 INX     H
```

```
022A CD 35 0D                 CALL    DCMP
022D 2B                       DCX     H          POINT TO COUNT CHAR AGAIN
022E DA 3B 02                 JC      CL2
0231 CA 3B 02                 JZ      CL2
0234               * INCREMENT TO NEXT LINE
0234 7E                       MOV     A,M
0235 CD C1 01                 CALL    ADR
0238 C3 26 02                 JMP     CL1
023B D5            CL2        PUSH    D
023C 11 D3 18                 LXI     D,IBUF     AREA TO UNPREPROCESS TO
023F CD 09 07                 CALL    UPPL
0242 23                       INX     H
0243 E5                       PUSH    H
0244 21 D3 18                 LXI     H,IBUF
0247 CD 1E 0D                 CALL    PRNTCR
024A CD 26 10                 CALL    PCHECK
024D CD 1C 10                 CALL    CRLF
0250 E1                       POP     H
0251 D1                       POP     D
0252 C3 26 02                 JMP     CL1
0255               * "RUN"
0255 CD DF 01      CRUN       CALL    CCLEAR
0258 2A 50 19                 LHLD    BOFA
025B 7E                       MOV     A,M
025C 3D                       DCR     A          CHECK FOR NULL PROGRAM
025D CA 62 00                 JZ      END
0260 23                       INX     H
0261 23                       INX     H
0262 23                       INX     H
0263 22 95 17                 SHLD    TXA
0266 22 B1 18                 SHLD    RTXA       POINTER FOR 'READ' STATEMENT
0269 AF                       XRA     A
026A 32 94 17                 STA     DIRF       CLEAR DIRECT FLAG AND FALL THROUGH TO DRIVER
026D CD 1C 10                 CALL    CRLF
0270               *
0270               *      INTERPRETER DRIVER
0270               *
0270 CD 26 10      ILOOP      CALL    PCHECK
0273 CD 7F 02                 CALL    ISTAT      INTERPRET CURRENT STATEMENT
0276 CD 2E 0C                 CALL    JOE        TEST FOR JUNK ON END
0279 D2 70 02                 JNC     ILOOP      CONTINUE IF NOT AT END OF PROGRAM
027C C3 62 00                 JMP     END        EXECUTE END STATEMENT
027F               * INTERPRET STATEMENT LOCATED BY TXA
027F CD 02 0D      ISTAT      CALL    GC         GET FIRST NON BLANK
0282 B7                       ORA     A
0283 F2 9C 02                 JP      LET        MUST BE LET IF NOT RW
0286 FE 91                    CPI     IRWLIM     IS IT AN INITIAL RW
0288 D2 AA 00                 JNC     E1
028B 11 E0 07                 LXI     D,STATD    STATEMENT DISPATCH TABLE BASE
028E CD 0A 0D      ISTA1      CALL    GCI        ADVANCE TEXT POINTER
0291 E6 1F                    ANI     37Q
0293 07                       RLC     .          MULTIPLY BY TWO PREPARING FOR TABLE LOOKUP
0294 6F                       MOV     L,A
0295 26 00                    MVI     H,0
0297 19                       DAD     D
0298 CD 40 0D                 CALL    LHLI
```

```
029B E9          0391          PCHL     .          BRANCH TO STATEMENT OR COMMAND
029C             0392 *
029C             0393 *              STATEMENTS
029C             0394 *
029C             0395 * "LET"
029C CD EA 0B    0396 LET      CALL     VAR        CHECK FOR VARAIABLE
029F DA AA 00    0397          JC       E1
02A2 E5          0398          PUSH     H          SAVE VALUE ADDRFSS
02A3 06 F5       0399          MVI      B,EQRW
02A5 CD FA 0C    0400          CALL     EATC
02A8 CD D7 05    0401          CALL     EXPRB
02AB D1          0402          POP      D          DESTINATION ADDRESS
02AC CD BE 0C    0403          CALL     POPA1      COPY EXPR VALUE TO VARIABLE
02AF C9          0404          RET
02B0             0405 * "FOR"
02B0 CD 63 0D    0406 SFOR     CALL     DIRT
02B3 CD EA 0B    0407          CALL     VAR        CONTROL VARIABLE
02B6 DA AA 00    0408          JC       E1
02B9 E5          0409          PUSH     H          CONTROL VARIABLE VALUE ADDRESS
02BA 06 F5       0410          MVI      B,EQRW
02BC CD FA 0C    0411          CALL     EATC
02BF CD D7 05    0412          CALL     EXPRB      INITIAL VALUE
02C2 D1          0413          POP      D          VARIABLE VALUE ADDRESS
02C3 D5          0414          PUSH     D          SAVE
02C4 CD BE 0C    0415          CALL     POPA1      SET INITIAL VALUE
02C7 06 9E       0416          MVI      B,TORW     RW FOR 'TO'
02C9 CD FA 0C    0417          CALL     EATC
02CC CD D7 05    0418          CALL     EXPRB      LIMIT VALUE COMPUTATION
02CF CD 02 0D    0419          CALL     GC         CHECK NEXT CHARACTER FOR POSSIBLE STEP EXPRESION
02D2 FE 9F       0420          CPI      STEPRW
02D4 CA E0 02    0421          JZ       FOR1
02D7             0422 * USE STEP OF 1
02D7 11 87 07    0423          LXI      D,FPONE
02DA CD A8 0C    0424          CALL     PSHA1
02DD C3 E6 02    0425          JMP      FOR2
02E0             0426 * COMPUTE STEP VALUE
02E0 CD 0A 0D    0427 FOR1     CALL     GCI        EAT THE STEP RW
02E3 CD D7 05    0428          CALL     EXPRB      THE STEP VALUE
02E6             0429 * HERE THE STEP AND LIMIT ARE ON ARG STACK
02E6 11 FE FF    0430 FOR2     LXI      D,-2       PREPARE TO ALLOCATE 2 BYTES ON CONTROL STACK
02E9 CD CD 0C    0431          CALL     PSHCS      RETURNS ADDRESS OF THOSE 2 BYTES IN HL
02EC EB          0432          XCHG
02ED CD 2E 0C    0433          CALL     JOE        TEST FOR JUNK ON END
02F0 DA B6 00    0434          JC       E4         NO "FOR" STATEMENT AT END OF PROGRAM
02F3 EB          0435          XCHG     .          DE HAS LOOP TEXT ADDR, HL HAS CONTROL STACK ADDR
02F4 72          0436          MOV      M,D        HIGH ORDER TEXT ADDRESS BYTE
02F5 2B          0437          DCX      H
02F6 73          0438          MOV      M,E        LOW ORDER TEXT ADDRESS BYTE
02F7 11 FB FF    0439          LXI      D,-FPSIZ   ALLOCATE SPACE FOR LIMIT ON CONTROL STACK
02FA CD CD 0C    0440          CALL     PSHCS
02FD E5          0441          PUSH     H          ADDR ON CONTROL STACK FOR LIMIT
02FE 11 FB FF    0442          LXI      D,-FPSIZ   ALLOCATE SPACE FOR STEP ON CONTROL STACK
0301 CD CD 0C    0443          CALL     PSHCS
0304 CD BD 0C    0444          CALL     POPAS      COPY STEP VALUE TO CONTROL STACK
0307 D1          0445          POP      D          CONTROL STACK ADDR FOR LIMIT VALUE
0308 CD BE 0C    0446          CALL     POPA1      LIMIT VALUE TO CONTROL STACK
```

```
030B 11 FD FF            LXI    D,-3          ALOCATE SPACE FOR TEXT ADDRESS AND CS ENTRY
030E CD CD 0C            CALL   PSHCS
0311 D1                  POP    D             CONTROL VARIABLE ADDRESS
0312 72                  MOV    M,D           HIGH ORDER BYTE OF CONTROL VARIABLE ADDRESS
0313 2B                  DCX    H
0314 73                  MOV    M,E           LOW ORDER BYTE OF CONTROL VARIABLE ADDRESS
0315 2B                  DCX    H
0316 36 01               MVI    M,FTYPE       SET CONTROL STACK ENTRY TYPE FOR 'FOR'
0318 C3 7C 03            JMP    NEXT5         GO FINISH OFF CAREFULLY
031B               * "NEXT"
031B CD 63 0D      NEXT  CALL   DIRT
031E 2A B5 18            LHLD   CSTKA         CONTROL STACK ADDRESS
0321 7E                  MOV    A,M           STACK ENTRY TYPE BYTE
0322 3D                  DCR    A             MUST BE FOR TYPE ELSE ERROR
0323 C2 B6 00            JNZ    E4            IMPROPER NESTING ERROR
0326 23                  INX    H             CONTROL STACK POINTER TO CONTROL VARIABLE ADDRESS
0327 E5                  PUSH   H
0328 CD EA 0B            CALL   VAR           CHECK VARIABLE, IN CASE USER WANTS
032B DA 37 03            JC     NEXT1         SKIP CHECK IF VAR NOT THERE
032E EB                  XCHG
032F E1                  POP    H             CONTROL VARIABLE ADDRESS
0330 E5                  PUSH   H             SAVE IT AGAIN
0331 CD 35 0D            CALL   DCMP
0334 C2 B6 00            JNZ    E4            IMPROPER NESTING IF NOT THE SAME
0337 E1            NEXT1 POP    H             CONTROL VARIABLE ADDRESS
0338 E5                  PUSH   H
0339 E5                  PUSH   H
033A 11 06 00            LXI    D,FPSIZ+2-1  COMPUTE ADDRESS TO STEP VALUE
033D 19                  DAD    D
033E E3                  XTHL                 NOW ADDRESS TO VAR IN HL
033F CD 40 0D            CALL   LHLI          VARIABLE ADDRESS
0342 44                  MOV    B,H           COPY VAR ADDRESS TO BC
0343 4D                  MOV    C,L
0344 D1                  POP    D             STEP VALUE ADDRESS
0345 D5                  PUSH   D
0346 CD C8 13            CALL   FADD          DO INCREMENT
0349 E1                  POP    H             STEP VALUE
034A 2B                  DCX    H             POINT TO SIGN OF STEP VALUE
034B 7E                  MOV    A,M           SIGN 0=POS, 1=NEG
034C 11 06 00            LXI    D,FPSIZ+1
034F 19                  DAD    D             PUTS LIMIT ADDRESS IN HL
0350 EB                  XCHG
0351 E1                  POP    H             VARIABLE ADDRESS
0352 CD 40 0D            CALL   LHLI          GET ADDRESS
0355 D5                  PUSH   D             SAVE CONTROL STACK POINTER TO GET TEXT ADDRESS
0356 B7                  ORA    A             SET CONDITIONS BASED ON SIGN OF STEP VALUE
0357 CA 5B 03            JZ     NEXT2         REVERSE TEST ON NEGATIVE STEP VALUE
035A EB                  XCHG
035B 44            NEXT2 MOV    B,H           SET UP ARGS FOR COMPARE
035C 4D                  MOV    C,L
035D CD 64 09            CALL   RELOP         TEST <=
0360 D1                  POP    D             TEXT ADDRESS
0361 FA 6F 03            JM     NEXT3         STILL SMALLER?
0364 CA 6F 03            JZ     NEXT3         JUMP IF WANT TO CONTINUE LOOP
0367               * TERMINATE LOOP
0367 21 03 00            LXI    H,3           REMOVE CSTACK ENTRY
```

```
036A 19                     DAD     D
036B 22 B5 18               SHLD    CSTKA
036E C9                     RET
036F 13          NEXT3      INX     D           TEXT ADDRESS
0370 EB                     XCHG
0371 CD 40 0D               CALL    LHLI        GET TEXT ADDRESS IN HL
0374             * ITERATE, SKIPPING NORMAL JUNK ON END TEST AT ILOOP
0374 EB          NEXT4      XCHG    .           SAVE NEW TEXT ADDRESS IN DE
0375 CD 2E 0C               CALL    JOE
0378 EB                     XCHG
0379 22 95 17    NEXT6      SHLD    TXA
037C 21 70 02    NEXT5      LXI     H,ILOOP
037F E3                     XTHL
0380 C9                     RET     .           TO DISPATCHER SKIPPING JOE CALL THERE
0381             * "IF"
0381 06 01       SIF        MVI     B,1         SPECIFY PRINCIPAL OPERATOR IS RELATIONAL
0383 CD D9 05               CALL    EXPB1
0386 2A 1C 19               LHLD    ASTKA       ADDRESS OF BOLLEAN VALUE ON ARG STACK
0389 34                     INR     M           SETS ZERO CONDITION IF RELATIONAL WAS TRUE
038A F5                     PUSH    PSW         SAVE CONDITIONS TO TEST LATER
038B CD BD 0C               CALL    POPAS       REMOVE VALUE FROM ARG STACK COPY TO SELF
038E F1                     POP     PSW
038F C2 08 04               JNZ     REM         IF TEST FALSE TREAT REST OF STATEMENT AS REM
0392             * TEST SUCCEEDED
0392 06 9D                  MVI     B,THENRW
0394 CD FA 0C               CALL    EATC
0397 CD E0 0D               CALL    INTGER      CHECK IF LINE NUMBER IS DESIRED ACTION
039A DA 7F 02               JC      ISTAT
039D C3 AA 03               JMP     GOTO1
03A0             * "GOTO"
03A0 AF          SGOTO      XRA     A
03A1 32 94 17               STA     DIRF        CLEAR DIRECT STATEMENT FLAG
03A4 CD E0 0D               CALL    INTGER      RETURNS INTEGER IN HL IF LINE NUMBER PRESENT
03A7 DA AA 00               JC      E1          SYNTAX ERROR NO LINE NUMBER
03AA EB          GOTO1      XCHG    .           LN IN DE
03AB CD 6E 0D               CALL    FINDLN      RETURNS TEXT ADDRESS POINTS TO COUNT VALUE
03AE 23          GOTO2      INX     H
03AF 23                     INX     H
03B0 23                     INX     H           ADVANCE TEXT POINTER PAST LINE NUMBER AND COUNT
03B1 C3 74 03               JMP     NEXT4
03B4             * "GOSUB"
03B4 CD 63 0D    GOSUB      CALL    DIRT
03B7 11 FD FF               LXI     D,-3        CREATE CONTROL STACK ENTRY
03BA CD CD 0C               CALL    PSHCS
03BD E5                     PUSH    H           SAVE STACK ADDRESS
03BE CD E0 0D               CALL    INTGER
03C1 DA AA 00               JC      E1
03C4 EB                     XCHG    .           LINE NUMBER TO DE
03C5 CD 2E 0C               CALL    JOE
03C8 44                     MOV     B,H
03C9 4D                     MOV     C,L
03CA E1                     POP     H           STACK ADDRESS
03CB 70                     MOV     M,B         STACK RETURN ADDRESS RETURNED BY JOE
03CC 2B                     DCX     H
03CD 71                     MOV     M,C
03CE 2B                     DCX     H
```

```
03CF 36 02           MVI   M,GTYPE    MAKE CONTROL STACK ENTRY TYPE 'GOSUB'
03D1 CD 6E 0D        CALL  FINDLN
03D4 23              INX   H
03D5 23              INX   H
03D6 23              INX   H
03D7 C3 79 03        JMP   NEXT6
03DA           * "RETURN"
03DA CD 63 0D  RETRN CALL  DIRT
03DD 32 94 17        STA   DIRF       CLEARS DIRF IF ACC IS CLEAR
03E0 2A B5 18        LHLD  CSTKA
03E3 7E        RET1  MOV   A,M
03E4 B7              ORA   A          CHECK FOR STACK EMPTY
03E5 CA B6 00        JZ    E4
03E8 FE 02           CPI   GTYPE      CHECK FOR GOSUB TYPE
03EA CA F4 03        JZ    RET2
03ED           * REMOVE FOR TYPE ENTRY FROM STACK
03ED 11 0F 00        LXI   D,FORSZ
03F0 19              DAD   D
03F1 C3 E3 03        JMP   RET1
03F4           * FOUND A GTYPE STACK ENTRY
03F4 23        RET2  INX   H
03F5 5E              MOV   E,M        LOW ORDER TEXT ADDRESS
03F6 23              INX   H
03F7 56              MOV   D,M        HIGH ORDER TEXT ADDRESS
03F8 23              INX   H          ADDRESS OF PREVIOUS CONTROL STACK ENTRY
03F9 22 B5 18        SHLD  CSTKA
03FC EB              XCHG  .          PUT TEXT ADDRESS IN HL
03FD 7E              MOV   A,M        ADDRESS POINTS TO EOF IF GOSUB WAS LAST LINE
03FE 3D              DCR   A          END OF FILE?
03FF C2 74 03        JNZ   NEXT4
0402 C3 62 00        JMP   END
0405           * "DATA AND REM"
0405 CD 63 0D  DATA  CALL  DIRT       DATA STATEMENT ILLEGAL AS DIRECT
0408 CD 0A 0D  REM   CALL  GCI
040B FE 0D           CPI   CR
040D C2 08 04        JNZ   REM
0410 2B              DCX   H          BACKUP POINTER SO NORMAL JOE WILL WORK
0411 22 95 17        SHLD  TXA
0414 C9              RET
0415           * "DIMENSION"
0415 CD 49 0C  DIM   CALL  NAME       LOOK FOR VARIABLE NAME
0418 DA AA 00        JC    E1
041B 79              MOV   A,C        PREPARE TURN ON 200Q BIT TO SIGNIFY MATRIX
041C F6 80           ORI   200Q
041E 4F              MOV   C,A
041F CD 58 0C        CALL  STLK
0422 D2 C2 00        JNC   E6         ERROR IF NAME ALREADY EXISTS
0425 E5              PUSH  H          SYMBOL TABLE ADDRESS
0426 06 E0           MVI   B,LPARRW
0428 CD FA 0C        CALL  EATC
042B CD D7 05        CALL  EXPRB
042E 06 29           MVI   B,')'
0430 CD FA 0C        CALL  EATC
0433 CD 8A 0D        CALL  PFIX       RETURN INTEGER IN DE
0436 21 AC 07        LXI   H,MATUB    MAXIMUM SIZE FOR MATRIX
0439 CD 35 0D        CALL  DCMP
```

```
043C D2 C2 00          JNC     E6
043F E1                POP     H          SYMBOL TABLE ADDRESS
0440 CD C6 0B          CALL    DIMS
0443 CD 02 0D          CALL    GC         SEE IF MORE TO DO
0446 FE 2C             CPI     ','
0448 C0                RNZ
0449 CD 0A 0D          CALL    GCI        EAT THE COMMA
044C C3 15 04          JMP     DIM
044F              * "STOP"
044F CD 63 0D     STOP    CALL    DIRT
0452 CD 19 10     STOP1   CALL    CRLF2
0455 32 3F 19             STA     BRKCHR
0458 21 65 07             LXI     H,STOPS
045B C3 D4 00             JMP     ERM1
045E              * "END"
045E              END     EQU     CMND1
045E              * "READ"
045E CD 63 0D     READ    CALL    DIRT
0461 2A 95 17             LHLD    TXA
0464 E5                   PUSH    H          SAVE TXA TEMPORARILY
0465 2A B1 18             LHLD    RTXA       THE 'READ' TXA
0468 22 95 17     READ0   SHLD    TXA
046B CD 0A 0D             CALL    GCI
046E FE 2C                CPI     ','
0470 CA 8B 04             JZ      READ2      PROCESS INPUT VALUE
0473 FE 87                CPI     DATARW
0475 CA 8B 04             JZ      READ2
0478 3D                   DCR     A
0479 CA B8 04             JZ      READ4
047C              * SKIP TO NEXT LINE
047C CD 08 04             CALL    REM        LEAVES ADDRESS OF LAST CR IN HL
047F 23                   INX     H
0480 7E                   MOV     A,M
0481 3D                   DCR     A
0482 CA B8 04             JZ      READ4
0485 23                   INX     H
0486 23                   INX     H
0487 23                   INX     H          HL NOW POINTS TO FIRST BYTE OF NEXT LINE
0488 C3 68 04             JMP     READ0
048B              * PROCESS VALUE
048B CD D7 05     READ2   CALL    EXPRB
048E CD 02 0D             CALL    GC
0491 FE 2C                CPI     ','        SKIP JOE TEST IF COMMA
0493 CA 99 04             JZ      READ3
0496              * JUNK ON END TEST
0496 CD 2E 0C             CALL    JOE
0499 2A 95 17     READ3   LHLD    TXA
049C 22 B1 18             SHLD    RTXA       SAVE NEW "READ" TEXT ADDRESS
049F E1                   POP     H          REAL TXA
04A0 22 95 17             SHLD    TXA
04A3 CD EA 0B             CALL    VAR
04A6 DA AA 00             JC      E1
04A9 CD BD 0C             CALL    POPAS      PUT READ VALUE INTO VARIABLE
04AC CD 02 0D             CALL    GC
04AF FE 2C                CPI     ','        CHECK FOR ANOTHER VARIABLE
04B1 C0                   RNZ
```

```
04B2 CD 0A 0D       0671          CALL    GCI         EAT THE COMMA
04B5 C3 5E 04       0672          JMP     READ
04B8 E1             0673 READ4    POP     H           PROGRAM TXA
04B9 22 95 17       0674          SHLD    TXA
04BC 01 44 52       0675          LXI     B,'RD'
04BF C3 C5 00       0676          JMP     ERROR
04C2                0677 *     "RESTORE"
04C2 2A 50 19       0678 RESTOR   LHLD    BOFA        BEGINNING OF FILE POINTER
04C5 23             0679          INX     H
04C6 23             0680          INX     H
04C7 23             0681          INX     H
04C8 22 B1 18       0682          SHLD    RTXA
04CB C9             0683          RET
04CC                0684 * "PRINT"
04CC CD 02 0D       0685 PRINT    CALL    GC
04CF FE 0D          0686          CPI     CR          CHECK FOR STAND ALONE PRINT
04D1 CA 1C 10       0687          JZ      CRLF
04D4 FE 22          0688 PRIN2    CPI     '"'
04D6 CA 16 05       0689          JZ      PSTR        PRINT THE STRING
04D9 FE 9C          0690          CPI     TABRW
04DB CA 58 05       0691          JZ      PTAB        TABULATION
04DE FE 25          0692          CPI     '%'
04E0 CA 23 05       0693          JZ      PFORM       SET FORMAT
04E3 FE 0D          0694          CPI     CR
04E5 C8             0695          RZ
04E6 FE 3B          0696          CPI     ';'
04E8 C8             0697          RZ
04E9 CD D7 05       0698          CALL    EXPRB       MUST BE EXPRESSION TO PRINT
04EC 11 BB 18       0699          LXI     D,FPSINK
04EF CD BE 0C       0700          CALL    POPA1       POP VALUE TO FPSINK
04F2 3A 90 17       0701          LDA     PHEAD
04F5 FE 38          0702          CPI     56
04F7 D4 1C 10       0703          CNC     CRLF        DO CRLF IF PRINT HEAD IS PAST 56
04FA 21 BB 18       0704          LXI     H,FPSINK
04FD CD 73 10       0705          CALL    FPOUT
0500 06 20          0706          MVI     B,' '
0502 CD 95 0E       0707          CALL    CHOUT
0505 CD 02 0D       0708 PR1      CALL    GC          GET DELIMITER
0508 FE 2C          0709          CPI     ','
050A C2 1C 10       0710          JNZ     CRLF
050D CD 0A 0D       0711 PRO      CALL    GCI
0510 CD 02 0D       0712          CALL    GC
0513 C3 D4 04       0713          JMP     PRIN2
0516 CD 0A 0D       0714 PSTR     CALL    GCI         GOBBLE THE QUOTE
0519 CD 23 0D       0715          CALL    PRNT        PRINT UP TO DOUBLE QUOTE
051C 23             0716          INX     H           MOVE POINTER PAST DOUBLE QUOTE
051D 22 95 17       0717          SHLD    TXA
0520 C3 05 05       0718          JMP     PR1
0523 3E 0C          0719 PFORM    MVI     A,2*FPNIB
0525 32 3A 19       0720          STA     INFES
0528 CD 0A 0D       0721          CALL    GCI         GOBBLE PREVIOUS CHAR
052B CD 0A 0D       0722 PFRM1    CALL    GCI
052E 21 3A 19       0723          LXI     H,INFES
0531 FE 25          0724          CPI     '%'         DELIMITER
0533 CA 05 05       0725          JZ      PR1
0536 06 80          0726          MVI     B,200Q
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
0538 FE 5A                  CPI     'Z'         TRAILING ZEROS?
053A CA 52 05               JZ      PF1
053D 06 01                  MVI     B,1
053F FE 45                  CPI     'E'         SCIENTIFIC NOTATION?
0541 CA 52 05               JZ      PF1
0544 CD FA 12               CALL    NMCHK
0547 D2 AA 00               JNC     E1
054A D6 30                  SUI     '0'         NUMBER OF DECIMAL PLACES
054C 07                     RLC
054D 47                     MOV     B,A
054E 7E                     MOV     A,M
054F E6 C1                  ANI     301Q
0551 77                     MOV     M,A
0552 7E             PF1     MOV     A,M
0553 B0                     ORA     B
0554 77                     MOV     M,A
0555 C3 2B 05               JMP     PFRM1
0558 CD 0A 0D       PTAB    CALL    GCI         GOBBLE TAB RW
055B 06 E0                  MVI     B,LPARRW
055D CD FA 0C               CALL    EATC
0560 CD D7 05               CALL    EXPRB
0563 06 29                  MVI     B,')'
0565 CD FA 0C               CALL    EATC
0568 CD 8A 0D               CALL    PFIX
056B 3A 90 17       PTAB1   LDA     PHEAD
056E BB                     CMP     E
056F D2 05 05               JNC     PR1
0572 06 20                  MVI     B,' '
0574 CD 95 0E               CALL    CHOUT
0577 C3 6B 05               JMP     PTAB1
057A                * "INPUT"
057A CD 02 0D       INPUT   CALL    GC
057D FE 2C                  CPI     ','
057F CA CB 05               JZ      NCRLF
0582 CD 1C 10               CALL    CRLF
0585 06 3F          INPO    MVI     B,'?'
0587 CD 95 0E               CALL    CHOUT
058A CD 55 0E       LINP    CALL    INLINE
058D 11 D3 18               LXI     D,IBUF
0590 D5             IN1     PUSH    D           SAVE FOR FPIN
0591 CD EA 0B               CALL    VAR
0594 DA AA 00               JC      E1
0597 D1                     POP     D
0598 06 00                  MVI     B,0
059A 1A                     LDAX    D
059B FE 2B                  CPI     '+'         LOOK FOR LEADING PLUS OR MINUS ON INPUT
059D CA A7 05               JZ      IN2
05A0 FE 2D                  CPI     '-'
05A2 C2 A8 05               JNZ     IN3
05A5 06 01                  MVI     B,1
05A7 13             IN2     INX     D
05A8 C5             IN3     PUSH    B
05A9 E5                     PUSH    H
05AA CD 57 12               CALL    FPIN        INPUT FP NUMBER
05AD DA D1 05               JC      INERR
05B0 E1                     POP     H
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
05B1 2B                    DCX     H
05B2 F1                    POP     PSW
05B3 77                    MOV     M,A
05B4 CD 02 0D              CALL    GC
05B7 FE 2C                 CPI     ','
05B9 C0                    RNZ     .          DONE IF NO MORE
05BA CD 0A 0D              CALL    GCI        EAT THE COMMA
05BD 78                    MOV     A,B        GET THE TERMINATOR TO A
05BE FE 2C                 CPI     ','
05C0 CA 90 05              JZ      IN1        GET THE NEXT INPUT VALUE FROM STRING
05C3              * GET NEW LINE FROM USER
05C3 06 3F                 MVI     B,'?'
05C5 CD 95 0E              CALL    CHOUT
05C8 C3 85 05              JMP     INPO
05CB CD 0A 0D     NCRLF    CALL    GCI
05CE C3 8A 05              JMP     LINP       NOW GET LINE
05D1 01 4E 49     INERR    LXI     B,'IN'
05D4 C3 C5 00              JMP     ERROR
05D7              *
05D7              *        EVALUATE AN EXPRESSION FROM TEXT
05D7              * HL TAKE OP TABLE ADDR OF PREVIOUS OPERATOR (NOT CHANGED)
05D7              * RESULT VALUE LEFT ON TOP OF ARG STACK, ARGF LEFT TRUE
05D7              *
05D7 06 00        EXPRB    MVI     B,0
05D9 21 EE 08     EXPB1    LXI     H,OPBOL
05DC AF                    XRA     A
05DD 32 91 17              STA     RELTYP
05E0              * ZERO IN B MEANS PRINCIPAL OPERATOR MAY NOT BE RELATIONAL
05E0 C5           EXPR     PUSH    B
05E1 E5                    PUSH    H          PUSH OPTBA
05E2 AF                    XRA     A
05E3 32 93 17              STA     ARGF
05E6 3A 93 17     EXPR1    LDA     ARGF
05E9 B7                    ORA     A
05EA C2 07 06              JNZ     EXPR2
05ED CD EA 0B              CALL    VAR        LOOK FOR VARIABLE PERHAPS SUBSCRIPTED
05F0 D4 A7 0C              CNC     PSHAS
05F3 D2 07 06              JNC     EXPR2
05F6 CD 47 0D              CALL    CONST
05F9 D2 07 06              JNC     EXPR2
05FC CD 02 0D              CALL    GC
05FF FE E0                 CPI     LPARRW
0601 21 D6 08              LXI     H,OPLPAR
0604 CA 83 06              JZ      XLPAR
0607              * ISN'T OR SHOULDN'T BE AN ARGUMENT
0607 CD 02 0D     EXPR2    CALL    GC
060A FE E0                 CPI     340Q       CHECK FOR RESERVED WORD OPERATOR
060C D2 25 06              JNC     XOP
060F FE C0                 CPI     300Q       CHECK FOR BUILT IN FUNCTION
0611 D2 74 06              JNC     XBILT
0614              * ILLEGAL EXPRESSION CHARACTER
0614 E1                    POP     H          GET OPTABA
0615 3A 93 17              LDA     ARGF
0618 B7                    ORA     A
0619 CA AA 00              JZ      E1
061C F1           XDON1    POP     PSW
```

```
061D 21 91 17          LXI    H,RELTYP   CHECK IF LEGAL PRINCIPAL OPERATOR
0620 BE                CMP    M
0621 C8                RZ
0622 C3 AA 00          JMP    E1
0625 E6 1F       XOP   ANI    37Q        CLEANS OFF RW BITS
0627 2A 93 17          LHLD   ARGF       TEST FOR ARGF TRUE
062A 2D                DCR    L
062B CA 40 06          JZ     XOP1
062E             * ARGF WAS FALSE, UNARY OPS ONLY POSSIBILITY
062E FE 05             CPI    '-'-OPBASE
0630 CA 3E 06          JZ     XOPM
0633 FE 03             CPI    '+'-OPBASE
0635 C2 AA 00          JNZ    E1
0638 CD 0A 0D          CALL   GCI        EAT THE '+'
063B C3 E6 05          JMP    EXPR1
063E 3E 09       XOPM  MVI    A,UMINUS-OPBASE
0640 CD 95 06    XOP1  CALL   OPADR
0643 D1                POP    D          PREVIOUS OPTBA
0644 1A                LDAX   D
0645 BE                CMP    M
0646 D2 1C 06          JNC    XDON1      NON-INCREASING PRECEDENCE
0649             * INCREASING PRECEDENCE CASE
0649 D5                PUSH   D          SAVE PREVIOUS OPTBA
064A E5                PUSH   H          SAVE CURRENT OPTBA
064B CD 0A 0D          CALL   GCI        TO GOBBLE OPERATOR
064E E1                POP    H
064F E5                PUSH   H
0650 06 00             MVI    B,0        SPECIFY NON-RELATIONAL
0652 CD E0 05          CALL   EXPR
0655 E1                POP    H
0656             * HL HAS OPTBA ADDRESS
0656             * SET UP ARGS AND PERFORM OPERATION ACTION
0656 E5          XOP2  PUSH   H
0657 7E                MOV    A,M
0658 2A 1C 19          LHLD   ASTKA
065B 44                MOV    B,H
065C 4D                MOV    C,L
065D E6 01             ANI    1
065F C2 6B 06          JNZ    XOP21
0662             * DECREMENT STACK POINTER BY ONE VALUE BINARY CASE
0662 11 05 00          LXI    D,FPSIZ
0665 19                DAD    D
0666 22 1C 19          SHLD   ASTKA
0669 54                MOV    D,H
066A 5D                MOV    E,L
066B 21 E6 05    XOP21 LXI    H,EXPR1
066E E3                XTHL   .          CHANGE RETURN LINK
066F 23                INX    H          SKIP OVER PRECIDENCE
0670 CD 40 0D          CALL   LHLI       LOAD ACTION ADDRESS
0673 E9                PCHL
0674             *
0674             *    ACTION ROUTINE CONVENTION
0674             * DE LEFT ARG AND RESULT FOR BINARY
0674             * BC RIGHT ARG FOR BINARY, ARG AND RESULT FOR UNARY
0674             * BUILT IN FUNCTION PROCESSING
0674             *
```

```
0674 CD 0A 0D     XBILT    CALL     GCI          EAT TOKEN
0677 E6 3F                 ANI      77Q          CLEAN OFF RW BITS
0679 2A 93 17              LHLD     ARGF         BUILT IN FUNCTION MUST COME AFTER OPERATOR
067C 2D                    DCR      L
067D CA AA 00              JZ       E1
0680 CD 95 06              CALL     OPADR        OPTBA TO HL
0683 E5           XLPAR    PUSH     H
0684 06 E0                 MVI      B,LPARRW
0686 CD FA 0C              CALL     EATC
0689 CD D7 05              CALL     EXPRB
068C 06 29                 MVI      B,')'
068E CD FA 0C              CALL     EATC
0691 E1                    POP      H            CODE FOR BUILT-IN FUNTION
0692 C3 56 06              JMP      XOP2
0695              * COMPUTE OPTABLE ADDRESS FOR OPERATOR IN ACC
0695 4F           OPADR    MOV      C,A
0696 06 00                 MVI      B,0
0698 21 D6 08              LXI      H,OPTAB
069B 09                    DAD      B
069C 09                    DAD      B
069D 09                    DAD      B            OPTAB ENTRY ADDR IS 3*OP+BASE
069E C9                    RET
069F              *
069F              * PREPROCESSOR, UN-PREPROCESSOR
069F              * PREPROCESS LINE IN IBUF BACK INTO IBUF
069F              * SETS CARRY IF LINE HAS NO LINE NUMBER
069F              * LEAVES CORRECT LENGTH OF LINE AFTER PREPROCESSING IN IBCN
069F              * IF THERE IS A LINE NUMBER, IT IS LOCATED AT IBLN=IBUF-2
069F              * TXA IS CLOBBERED
069F              *
069F 21 D3 18     PP       LXI      H,IBUF       FIRST CHARACTER OF INPUT LINE
06A2 22 95 17              SHLD     TXA          SO GCI WILL WORK
06A5 CD E0 0D              CALL     INTGER       SETS CARRY IF NO LINE NUMBER
06A8 22 D1 18              SHLD     IBLN         STORE LINE NUMBER VALUE (EVEN IF NONE)
06AB F5                    PUSH     PSW          SAVE STATE OF CARRY BIT FOR RETURNING
06AC 2A 95 17              LHLD     TXA          ADDRESS OF NEXT CHARACTER IN IBUF
06AF 0E 04                 MVI      C,4          SET UP INITIAL VALUE FOR COUNT
06B1 11 D3 18              LXI      D,IBUF       INITIALIZE WRITE POINTER
06B4              * COME HERE TO CONTINUE PREPROCESSING LINE
06B4 D5           PPL      PUSH     D
06B5 11 02 08              LXI      D,RWT        BASE OF RWT
06B8 E5           PPL1     PUSH     H            SAVE TEXT ADDRESS
06B9 1A                    LDAX     D            RW VALUE FOR THIS ENTRY IN RWT
06BA 47                    MOV      B,A          SAVE IN B IN CASE OF MATCH
06BB 13           PPL2     INX      D            ADVANCE ENTRY POINTER TO NEXT BYTE
06BC 1A                    LDAX     D            GET NEXT CHARACTER FROM ENTRY
06BD BE                    CMP      M            COMPARE WITH CHARACTER IN TEXT
06BE C2 C5 06              JNZ      PPL3
06C1 23                    INX      H            ADVANCE TEXT POINTER
06C2 C3 BB 06              JMP      PPL2         CONTINUE COMPARISON
06C5              * COME HERE WHEN COMPARISON OF BYTE FAILED
06C5 B7           PPL3     ORA      A
06C6 FA F7 06              JM       PPL6         JUMP IF FOUND MATCH
06C9              * SCAN TO BEGINNING OF NEXT ENTRY
06C9 13           PPL4     INX      D            ADVANCE ENTRY POINTER
06CA 1A                    LDAX     D            NEXT BYTE IS EITHER CHARACTER OR RW BYTE
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
06CB B7                       ORA     A
06CC F2 C9 06                 JP      PPL4        KEEP SCANNING IF NOT RW BYTE
06CF               * NOW SEE IF AT END OF TABLE, AND FAIL OR RETURN CONDITION
06CF E1                       POP     H           RECOVER ORIGIONAL TEXT POINTER
06D0 EE FF                    XRI     377Q        CHECK FOR END OF TABLE BYTE
06D2 C2 B8 06                 JNZ     PPL1        CONTINUE SCAN OF TABLE
06D5               * DIDN'T FIND AN ENTRY AT THE GIVEN TEXT ADDR
06D5 D1                       POP     D
06D6 7E                       MOV     A,M         GET TEXT CHARACTER
06D7 FE 0D                    CPI     CR          CHECK FOR END OF LINE
06D9 CA 00 07                 JZ      PPL8        GO CLEAN UP AND RETURN
06DC 12                       STAX    D
06DD 13                       INX     D
06DE 0C                       INR     C
06DF 23                       INX     H           ADVANCE TEXT POINTER
06E0 FE 22                    CPI     '"'         CHECK FOR QUOTED STRING POSSIBILITY
06E2 C2 B4 06                 JNZ     PPL         RESTART RWT SEARCH AT NET CHARACTER POSITION
06E5               * HERE WE HAVE A QUOTED STRING, SO EAT TILL ENDQUOTE
06E5 7E            PPL5       MOV     A,M         NEXT CHARACTER
06E6 FE 0D                    CPI     CR
06E8 CA 00 07                 JZ      PPL8        NO STRING ENDQUOTE, LET INTERPRETER WORRY
06EB 12                       STAX    D
06EC 13                       INX     D
06ED 0C                       INR     C
06EE 23                       INX     H           ADVANCE TEXT POINTER
06EF FE 22                    CPI     '"'
06F1 CA B4 06                 JZ      PPL         BEGIN RWT SCAN FROM NEW CHARACTER POSITION
06F4 C3 E5 06                 JMP     PPL5
06F7               * FOUND MATCH SO PUT RW VALUE IN TEXT
06F7 F1            PPL6       POP     PSW         REMOVE UNNEEDED TEST POINTER FROM STACK
06F8 D1                       POP     D
06F9 78                       MOV     A,B
06FA 12                       STAX    D
06FB 13                       INX     D
06FC 0C                       INR     C
06FD C3 B4 06                 JMP     PPL
0700               * COME HERE WHEN DONE
0700 3E 0D         PPL8       MVI     A,CR
0702 12                       STAX    D
0703 21 D0 18                 LXI     H,IBCNT     SET UP COUNT IN CASE LINE OF LINE NUMBER
0706 71                       MOV     M,C
0707 F1                       POP     PSW         RESTORE CARRY CONDITION (LINE NUMBER FLAG)
0708 C9                       RET
0709               *
0709               * UN PREPROCESS LINE ADDRESSES IN HL TO DE BUFFER
0709               * RETURN SOURCE ADDRESS OF CR IN HL ON RETURN
0709               *
0709 23            UPPL       INX     H           SKIP OVER COUNT BYTE
070A E5                       PUSH    H           SAVE SOURCE TEXT POUNTER
070B CD 40 0D                 CALL    LHLI        LOAD LINE NUMBER VALUE
070E CD FA 0D                 CALL    CNS         CONVERT LINE NUMBER
0711 3E 20                    MVI     A,' '
0713 12                       STAX    D           PUT BLANK AFTER LINE NUMBER
0714 13                       INX     D           INCREMENT DESTINATION POINTER
0715 E1                       POP     H
0716 23                       INX     H           INCREMENT H PAST LINE NUMBER
```

```
0717 23           1007 UPPO     INX      H
0718 7E           1008          MOV      A,M        NEXT TOKEN IN SOURCE
0719 B7           1009          ORA      A
071A FA 25 07     1010          JM       UPP1       JUMP IF TOKEN IS RW
071D 12           1011          STAX     D          PUT CHARACTER IN BUFFER
071E FE 0D        1012          CPI      CR         CHECK FOR DONE
0720 C8           1013          RZ
0721 13           1014          INX      D          ADVANCE DESTINATION BUFFER ADDRESS
0722 C3 17 07     1015          JMP      UPPO
0725              1016 * COME HERE WHEN RW BYTE DETECTED IN SOURCE
0725 E5           1017 UPP1     PUSH     H          SAVE SOURCE POINTER
0726 21 02 08     1018          LXI      H,RWT      BASE OF RWT
0729 BE           1019 UPP2     CMP      M          SEE IF RW MATCHED RWT ENTRY
072A 23           1020          INX      H          ADVANCE RWT POINTER
072B C2 29 07     1021          JNZ      UPP2       CONTINUE LOOKING IF NOT FOUND
072E              1022 * FOUND MATCH, ENTRY POINTER LOCATES FIRST CHARACTER
072E 7E           1023 UPP3     MOV      A,M        CHARACTER OF RW
072F B7           1024          ORA      A          CHECK FOR DONE
0730 FA 39 07     1025          JM       UPP4
0733 12           1026          STAX     D
0734 13           1027          INX      D
0735 23           1028          INX      H
0736 C3 2E 07     1029          JMP      UPP3
0739              1030 * COME HERE IF DONE WITH RW TRANSFER
0739 E1           1031 UPP4     POP      H          SOURCE POINTER
073A C3 17 07     1032          JMP      UPPO
073D              1033 *
073D              1034 *        CONSTANTS AND TABLES
073D              1035 *
073D 52 45 41 44  1036 RDYS     ASC      'READY"'
     59 22
0743 50 52 4F 47  1037 PLS      ASC      'PROGRAM LOADED? "'
     52 41 4D 20
     4C 4F 41 44
     45 44 3F 20
     22
0754 20 45 52 52  1038 ERS      ASC      ' ERROR"'
     4F 52 22
075B 20 49 4E 20  1039 INS      ASC      ' IN LINE "'
     4C 49 4E 45
     20 22
0765 53 54 4F 50  1040 STOPS    ASC      'STOP"'
     22
076A 46 49 52 53  1041 FAS      ASC      'FIRST ADDR "'
     54 20 41 44
     44 52 20 22
0776 4C 41 53 54  1042 LAS      ASC      'LAST ADDR "'
     20 41 44 44
     52 20 22
0781              1043 *
0781 FF           1044          DB       -1         FLAGS END OF SINE COEFFICIENT LIST
0782 00           1045          DB       0
0783 10           1046          DB       1*16
0784 00 00        1047          DW       0
0786 00           1048          DB       0
0787 81           1049 FPONE    DB       129        EXPONENT
```

```
0788            1050 *           SINE COEFFICIENT LIST
0788            1051 * NOTE: THE FLOATING PNT 1 ABOVE IS A PART OF THIS TABLE
0788 16         1052       DB        1*16+6
0789 66         1053       DB        6*16+6
078A 67         1054       DB        6*16+7
078B 01         1055       DB        1
078C 80         1056       DB        128          -.166667 E 0 (-1/3!)
078D 83         1057       DB        8*16+3
078E 33         1058       DB        3*16+3
078F 33         1059       DB        3*16+3
0790 00         1060       DB        0
0791 7E         1061       DB        128-2        .833333 E-2 (1/5!)
0792 19         1062       DB        1*16+9
0793 84         1063       DB        8*16+4
0794 13         1064       DB        1*16+3
0795 01         1065       DB        1
0796 7D         1066       DB        128-3        -.198413 E-3 (-1/7!)
0797 27         1067       DB        2*16+7
0798 55         1068       DB        5*16+5
0799 73         1069       DB        7*16+3
079A 00         1070       DB        0
079B 7B         1071       DB        128-5        .275573 E-5 (1/9!)
079C 25         1072       DB        2*16+5
079D 05         1073       DB        0*16+5
079E 21         1074       DB        2*16+1
079F 01         1075       DB        1
07A0 79         1076 SINX  DB        128-7        -.250521 E-7 (-1/11!)
07A1            1077 *           COSINE COEFFICIENT LIST
07A1 FF         1078       DB        -1           MARKS END OF LIST
07A2 00         1079       DB        0
07A3 10         1080       DB        1*16+0
07A4 00         1081       DB        0
07A5 00         1082       DB        0
07A6 00         1083       DB        0
07A7 81         1084       DB        128+1        .100000 E 1 (1/1!)
07A8 50         1085       DB        5*16+0
07A9 00         1086       DB        0
07AA 00         1087       DB        0
07AB 01         1088       DB        1
07AC 80         1089 MATUB DB        128          -.500000 E 0 (-1/2!)
07AD 41         1090       DB        4*16+1
07AE 66         1091       DB        6*16+6
07AF 67         1092       DB        6*16+7
07B0 00         1093       DB        0
07B1 7F         1094 RANDS DB        128-1        .416667 E-1 (1/4!)
07B2 13         1095       DB        1*16+3
07B3 88         1096       DB        8*16+8
07B4 89         1097       DB        8*16+9
07B5 01         1098       DB        1
07B6 7E         1099       DB        128-2        .138889 E-2 (-1/6!)
07B7 24         1100       DB        2*16+4
07B8 80         1101       DB        8*16+0
07B9 16         1102       DB        1*16+6
07BA 00         1103       DB        0
07BB 7C         1104       DB        128-4        .248016 E-4 (1/8!)
07BC 27         1105       DB        2*16+7
```

```
07BD 55          1106          DB      5*16+5
07BE 73          1107          DB      7*16+3
07BF 01          1108          DB      1
07C0 7A          1109 COSX     DB      128-6       .275573 E-6 (-1/101)
07C1 20          1110          DB      2*16
07C2 00 00       1111          DW      0
07C4 00          1112          DB      0
07C5 81          1113 FPTWO    DB      129
07C6 15          1114          DB      1*16+5
07C7 70          1115          DB      7*16+0
07C8 80          1116          DB      8*16+0
07C9 00          1117          DB      0
07CA 81          1118 PIC2     DB      128+1       PI/2 .157080 E 1
07CB 63          1119          DB      6*16+3
07CC 66          1120          DB      6*16+6
07CD 20          1121          DB      2*16+0
07CE 00          1122          DB      0
07CF 80          1123 PIC1     DB      128         2/PI .636620 E 0
07D0 97 17       1124 LCSTKA   DW      CSTKL
07D2             1125 *
07D2             1126 *            COMMAND TABLE
07D2             1127 *
07D2 55 02       1128 CMNDD    DW      CRUN        0
07D4 15 02       1129          DW      CLIST       1
07D6 08 02       1130          DW      CNULL       2
07D8 D7 01       1131          DW      CSCR        3
07DA 00 00       1132          DW      START       4 SET UP MEMORY BOUNDS
07DC 00 00     U 1133          DW      TSAV        5 TAPE SAVE JUMP ADDRESS POINTER
07DE 00 00     U 1134          DW      TLOAD       6 TAPE LOAD
07E0             1135 *  STATEMENT TABLE
07E0 9C 02       1136 STATD    DW      LET         0
07E2 1B 03       1137          DW      NEXT        1
07E4 81 03       1138          DW      SIF         2
07E6 A0 03       1139          DW      SGOTO       3
07E8 B4 03       1140          DW      GOSUB       4
07EA DA 03       1141          DW      RETRN       5
07EC 5E 04       1142          DW      READ        6
07EE 05 04       1143          DW      DATA        7
07F0 B0 02       1144          DW      SFOR        10
07F2 CC 04       1145          DW      PRINT       11
07F4 7A 05       1146          DW      INPUT       12
07F6 15 04       1147          DW      DIM         13
07F8 4F 04       1148          DW      STOP        14
07FA 62 00       1149          DW      END         15
07FC C2 04       1150          DW      RESTOR      16
07FE 08 04       1151          DW      REM         17
0800 DF 01       1152          DW      CCLEAR      20
0802             1153 *
0802             1154 * R/W WORD TABLE FORMAT IS RESERVED WORD FOLLOWED BY CHR
0802             1155 * OF RESERVED WORD. LAST ENTRY IS FOLLOWED BY A 377Q
0802             1156 * RW'S THAT ARE SUBSTRINGS OF OTHER RW'S (E.G. >) MUST
0802             1157 * FOLLOW THE LARGER WORD.
0802             1158 *
0802 80          1159 RWT      DB      200Q
0803 4C 45 54    1160          ASC     "LET"
0806 81          1161          DB      201Q
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
0807 4E 45 58 54               ASC    "NEXT"
080B 82                        DB     202Q
080C 49 46                     ASC    "IF"
080E 83                        DB     203Q
080F 47 4F 54 4F               ASC    "GOTO"
0813 84                        DB     204Q
0814 47 4F 53 55               ASC    "GOSUB"
     42
0819 85                        DB     205Q
081A 52 45 54 55               ASC    "RETURN"
     52 4E
0820 86                        DB     206Q
0821 52 45 41 44               ASC    "READ"
0825 87                        DB     207Q
0826 44 41 54 41               ASC    'DATA'
082A                   DATARW  EQU    207Q
082A 88                        DB     210Q
082B 46 4F 52                  ASC    "FOR"
082E 89                        DB     211Q
082F 50 52 49 4E               ASC    "PRINT"
     54
0834 89                        DB     211Q
0835 3A                        DB     ':'
0836 8A                        DB     212Q
0837 49 4E 50 55               ASC    "INPUT"
     54
083C 8B                        DB     213Q
083D 44 49 4D                  ASC    'DIM'
0840 8C                        DB     214Q
0841 53 54 4F 50               ASC    'STOP'
0845 8D                        DB     215Q
0846 45 4E 44                  ASC    'END'
0849 8E                        DB     216Q
084A 52 45 53 54               ASC    'RESTORE'
     4F 52 45
0851 8F                        DB     217Q
0852 52 45 4D                  ASC    'REM'
0855 90                        DB     220Q
0856 43 4C 45 41               ASC    'CLEAR'
     52
085B                   CLRRW   EQU    220Q
085B                   IRWLIN  EQU    221Q       LAST INITIAL RESERVED WORD VALUE + 1
085B 9F                        DB     237Q
085C 53 54 45 50               ASC    'STEP'
0860                   STEPRW  EQU    237Q
0860 9E                        DB     236Q
0861 54 4F                     ASC    'TO'
0863                   TORW    EQU    236Q
0863 9D                        DB     235Q
0864 54 48 45 4E               ASC    'THEN'
0868                   THENRW  EQU    235Q
0868 9C                        DB     234Q
0869 54 41 42                  ASC    'TAB'
086C                   TABRW   EQU    234Q
086C A0                        DB     240Q
086D 52 55 4E                  ASC    'RUN'
```

```
0870                RUNRW   EQU  240Q
0870 A1                     DB   241Q
0871 4C 49 53 54            ASC  'LIST'
0875                LISTRW  EQU  241Q
0875 A2                     DB   242Q
0876 4E 55 4C 4C            ASC  'NULL'
087A                NULLRW  EQU  242Q
087A A3                     DB   243Q
087B 53 43 52               ASC  'SCR'
087E                SCRRW   EQU  243Q
087E A4                     DB   244Q
087F 4D 45 4D               ASC  'MEM'
0882                MEMRW   EQU  245Q
0882 A5                     DB   245Q
0883 54 53 41 56            ASC  "TSAV"
0887 A6                     DB   246Q
0888 54 4C 4F 41            ASC  "TLOAD"
     44
088D                LPARRW  EQU  '('-OPBASE+340Q
088D E0                     DB   LPARRW
088E 28                     DB   '('
088F E2                     DB   '*'-OPBASE+340Q
0890 2A                     DB   '*'
0891                PLSRW   EQU  '+'-OPBASE+340Q
0891 E3                     DB   PLSRW
0892 2B                     DB   '+'
0893                MINRW   EQU  '-'-OPBASE+340Q
0893 E5                     DB   MINRW
0894 2D                     DB   '-'
0895 E7                     DB   '/'-OPBASE+340Q
0896 2F                     DB   '/'
0897 EF                     DB   67Q-OPBASE+340Q
0898 3E 3D                  ASC  '>='
089A F0                     DB   70Q-OPBASE+340Q
089B 3C 3D                  ASC  '<='
089D F1                     DB   71Q-OPBASE+340Q
089E 3C 3E                  ASC  '<>'
08A0 EA                     DB   62Q-OPBASE+340Q
08A1 3D 3E                  ASC  '=>'
08A3 EB                     DB   63Q-OPBASE+340Q
08A4 3D 3C                  ASC  '=<'
08A6 F4                     DB   '<'-OPBASE+340Q
08A7 3C                     DB   '<'
08A8                EQRW    EQU  '='-OPBASE+340Q
08A8 F5                     DB   EQRW
08A9 3D                     DB   '='
08AA F6                     DB   '>'-OPBASE+340Q
08AB 3E                     DB   '>'
08AC C1                     DB   301Q
08AD 41 42 53               ASC  'ABS'
08B0 C6                     DB   306Q
08B1 49 4E 54               ASC  'INT'
08B4 CC                     DB   314Q
08B5 41 52 47               ASC  'ARG'
08B8 CD                     DB   315Q
08B9 43 41 4C 4C            ASC  'CALL'
```

```
08BD CE                       DB      316Q
08BE 52 4E 44                 ASC     'RND'
08C1 D2                       DB      322Q
08C2 53 47 4E                 ASC     'SGN'
08C5 D3                       DB      323Q
08C6 53 49 4E                 ASC     'SIN'
08C9 C4                       DB      304Q
08CA 53 51 52                 ASC     'SQR'
08CD D7                       DB      327Q
08CE 54 41 4E                 ASC     'TAN'
08D1 D8                       DB      330Q
08D2 43 4F 53                 ASC     'COS'
08D5 FF                       DB      377Q
08D6                *
08D6                *         OPERATION TABLE
08D6                *
08D6 OF             OPTAB     DB      15
08D7                OPLPAR    EQU     OPTAB
08D7 BC 09                    DW      ALPAR
08D9 OF                       DB      15
08DA CC 09                    DW      AABS
08DC 0A                       DB      10
08DD 9D 09                    DW      AMUL
08DF 06                       DB      6
08E0 89 09                    DW      AADD
08E2 OF                       DB      15
08E3 57 0A                    DW      ASQR
08E5 06                       DB      6
08E6 93 09                    DW      ASUB
08E8 OF                       DB      15
08E9 AA 0B                    DW      AINT
08EB 0A                       DB      10
08EC A7 09                    DW      ADIV
08EE 01             OPBOL     DB      1
08EF 00 00                    DW      0
08F1 0D                       DB      13
08F2 BD 09                    DW      ANEG
08F4 04                       DB      4
08F5 4C 09                    DW      AGE
08F7 04                       DB      4
08F8 58 09                    DW      ALE
08FA OF                       DB      15
08FB 79 0B                    DW      AARG
08FD OF                       DB      15
08FE 86 0B                    DW      ACALL
0900 OF                       DB      15
0901 AD 0A                    DW      ARND
0903 04                       DB      4
0904 4C 09                    DW      AGE
0906 04                       DB      4
0907 58 09                    DW      ALE
0909 04                       DB      4
090A 43 09                    DW      ANE
090C OF                       DB      15
090D D2 09                    DW      ASGN
090F OF                       DB      15
```

```
0910 ED 09             DW      ASIN
0912 04                DB      4
0913 2D 09             DW      ALT
0915 04                DB      4
0916 3A 09             DW      AEQ
0918 04                DB      4
0919 21 09             DW      AGT
091B 0F                DB      15
091C 3C 0A             DW      ATAN
091E 0F                DB      15
091F 30 0A             DW      ACOS
0921          *
0921          *      ACTION ROUTINES FOR RELATIONAL OPERATORS
0921          *
0921 CD 64 09 AGT      CALL    RELOP
0924 CA 2A 09          JZ      RFALSE
0927 FA 36 09          JM      RTRUE
092A AF       RFALSE   XRA     A
092B 12                STAX    D
092C C9                RET
092D CD 64 09 ALT      CALL    RELOP
0930 CA 2A 09          JZ      RFALSE
0933 FA 2A 09          JM      RFALSE
0936 3E FF    RTRUE    MVI     A,377Q
0938 12                STAX    D
0939 C9                RET
093A CD 64 09 AEQ      CALL    RELOP
093D CA 36 09          JZ      RTRUE
0940 C3 2A 09          JMP     RFALSE
0943 CD 64 09 ANE      CALL    RELOP
0946 CA 2A 09          JZ      RFALSE
0949 C3 36 09          JMP     RTRUE
094C CD 64 09 AGE      CALL    RELOP
094F CA 36 09          JZ      RTRUE
0952 FA 36 09          JM      RTRUE
0955 C3 2A 09          JMP     RFALSE
0958 CD 64 09 ALE      CALL    RELOP
095B CA 36 09          JZ      RTRUE
095E FA 2A 09          JM      RFALSE
0961 C3 36 09          JMP     RTRUE
0964          *     COMMON ROUTINE FOR RELATIONAL OPERATOR ACTION
0964          * LEFT ARG ADDR IN DE, SAVED
0964          * RIGHT ARG ADDR IN BC
0964          * ON RETURN SIGN SET=GT, ZERO SET=EQUAL
0964 D5       RELOP    PUSH    D
0965 0B                DCX     B
0966 1B                DCX     D
0967 60                MOV     H,B
0968 69                MOV     L,C
0969 1A                LDAX    D
096A 96                SUB     M
096B 23                INX     H
096C 13                INX     D
096D C2 82 09          JNZ     RLOP1      TEST SIGNS OF ARGS IF DIFFERENT THEN RET
0970 01 BB 18          LXI     B,FPSINK
0973 CD 1F 14          CALL    FSUB
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608

** COPYRIGHT 1976 **

```
0976 3A BB 18               LDA     FPSINK      CHECK FOR ZERO RESULT
0979 B7                     ORA     A
097A CA 82 09               JZ      RLOP1
097D 3A BA 18               LDA     FPSINK-1    SIGN OF FPSINK
0980 07                     RLC
0981 3D                     DCR     A
0982 3E 01          RLOP1   MVI     A,1
0984 32 91 17               STA     RELTYP      SET RELTYP TRUE
0987 D1                     POP     D
0988 C9                     RET
0989                *
0989                *   ACTION ROUTINES FOR ARITHMETIC OPERATORS
0989                *           (CODE WASTERS)
0989 60             AADD    MOV     H,B
098A 69                     MOV     L,C
098B 42                     MOV     B,D
098C 4B                     MOV     C,E
098D CD C8 13       AADD1   CALL    FADD
0990 C3 AE 09               JMP     FPETST
0993 60             ASUB    MOV     H,B
0994 69                     MOV     L,C
0995 42                     MOV     B,D
0996 4B                     MOV     C,E
0997 CD 1F 14       ASUB1   CALL    FSUB
099A C3 AE 09               JMP     FPETST
099D 60             AMUL    MOV     H,B
099E 69                     MOV     L,C
099F 42                     MOV     B,D
09A0 4B                     MOV     C,E
09A1 CD 98 14       AMUL1   CALL    FMUL
09A4 C3 AE 09               JMP     FPETST
09A7 60             ADIV    MOV     H,B
09A8 69                     MOV     L,C
09A9 42                     MOV     B,D
09AA 4B                     MOV     C,E
09AB CD A1 15       ADIV1   CALL    FDIV
09AE AF             FPETST  XRA     A
09AF 32 91 17               STA     RELTYP
09B2 3A 23 17               LDA     ERRI
09B5 B7                     ORA     A
09B6 C8                     RZ
09B7 2A 1C 19               LHLD    ASTKA       ZERO RESULT ON UNDERFLOW
09BA 36 00          FPET1   MVI     M,0
09BC C9             ALPAR   RET
09BD                *
09BD                *   UNARY AND BUILT IN FUNCTION ACTION ROUTINES
09BD                *
09BD 0A             ANEG    LDAX    B
09BE B7                     ORA     A
09BF CA C7 09               JZ      ANEG1
09C2 0B                     DCX     B
09C3 0A                     LDAX    B
09C4 EE 01                  XRI     1
09C6 02                     STAX    B
09C7 AF             ANEG1   XRA     A
09C8 32 91 17               STA     RELTYP
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
09CB C9                    RET
09CC 0B           AABS     DCX     B
09CD AF                    XRA     A
09CE 02                    STAX    B
09CF C3 C7 09              JMP     ANEG1
09D2 CD C7 09     ASGN     CALL    ANEG1
09D5 50                    MOV     D,B
09D6 59                    MOV     E,C
09D7 0A                    LDAX    B          GET EXPONENT
09D8 B7                    ORA     A
09D9 C2 DE 09              JNZ     ASGN1
09DC 12                    STAX    D          MAKE ARGUMENT ZERO
09DD C9                    RET
09DE 0B           ASGN1    DCX     B
09DF 0A                    LDAX    B
09E0 B7                    ORA     A
09E1 21 87 07              LXI     H,FPONE
09E4 CA 9C 0C              JZ      VCOPY
09E7 21 C7 13              LXI     H,FPNONE
09EA C3 9C 0C              JMP     VCOPY
09ED              *
09ED              *     COMPUTE SIN(X) X=TOP OF ARGUMENT STACK
09ED              *     RETURN RESULT IN PLACE OF X
09ED              *
09ED CD 31 0B     ASIN     CALL    QUADC      COMPUTE QUADRANT
09F0 2A 1C 19              LHLD    ASTKA
09F3 54                    MOV     D,H
09F4 5D                    MOV     E,L
09F5 01 C0 18              LXI     B,FTEMP
09F8 CD A1 09              CALL    AMUL1      FTEMP=X*X
09FB F1                    POP     PSW
09FC F5                    PUSH    PSW        A=QUADRANT
09FD 1F                    RAR
09FE DA 27 0A              JC      SIN10      QUAD ODD. COMPUTE COSINE
0A01              *   COMPUTE X*P(X*X) -- SINE
0A01 11 C5 18              LXI     D,FTEM1
0A04 2A 1C 19              LHLD    ASTKA
0A07 CD 9C 0C              CALL    VCOPY      FTEM1=X*X
0A0A 01 A0 07              LXI     B,SINX
0A0D CD 03 0B              CALL    POLY       P(X*X)
0A10 CD 2A 0B              CALL    PREPOP
0A13 21 C5 18              LXI     H,FTEM1
0A16 CD A1 09              CALL    AMUL1      X*P(X*X)
0A19              *     COMPUTE SIGN OF RESULT
0A19              * POSITIVE FOR QUADRANTS 0,1. NEGATIVE FOR 2,3
0A19              * NEGATE ABOVE FOR NEGATIVE ARGUMENTS
0A19 F1           SIN5     POP     PSW        QUADRANT
0A1A 47                    MOV     B,A
0A1B F1                    POP     PSW        SIGN
0A1C 07                    RLC     .          SIGN, 2 TO THE 1ST BIT
0A1D A8                    XRA     B          QUADRANT, MAYBE MODIFIED FOR NEGATIVE ARG.
0A1E 2A 1C 19              LHLD    ASTKA
0A21 2B                    DCX     H          PTR TO SIGN
0A22 D6 02                 SUI     2
0A24 F8                    RM      .          QUADRANT 0 OR 1
0A25 34                    INR     M          ELSE SET RESULT NEGATIVE
```

```
0A26 C9                          RET
0A27                   * COMPUTE P(X*X) -- COSINE
0A27 01 C0 07          SIN10     LXI      B,COSX
0A2A CD 03 0B                    CALL     POLY         P(X*X)
0A2D C3 19 0A                    JMP      SIN5
0A30                   *
0A30                   *     COMPUTE COS(X) X=TOP OF ARGUMENT STACK
0A30                   * RETURN RESULT IN PLACE OF X
0A30                   * COS(X)=SIN(X+PI/2)
0A30                   *
0A30 CD 2A 0B          ACOS      CALL     PREPOP
0A33 21 CA 07                    LXI      H,PIC2       PI/2
0A36 CD 8D 09                    CALL     AADD1        TOS=TOS+PI/2
0A39 C3 ED 09                    JMP      ASIN
0A3C                   *
0A3C                   *     COMPUTE TAN(X) X=TOP OF ARGUMENT STACK
0A3C                   * RETURN RESULT IN PLACE OF X
0A3C                   * TAN(X)=SIN(X)/COS(X)
0A3C                   *
0A3C 2A 1C 19          ATAN      LHLD     ASTKA
0A3F CD A7 0C                    CALL     PSHAS        PUSH COPY OF X ONTO ARG STACK
0A42 CD 30 0A                    CALL     ACOS         COS(X)
0A45 11 CA 18                    LXI      D,FTEM2
0A48 CD BE 0C                    CALL     POPA1        FTEM2=COS(X)
0A4B CD ED 09                    CALL     ASIN
0A4E CD 2A 0B                    CALL     PREPOP
0A51 21 CA 18                    LXI      H,FTEM2
0A54 C3 AB 09                    JMP      ADIV1        SIN(X)/COS(X)
0A57                   *
0A57                   *     COMPUTE SQR(X) X=TOP OF ARGUMENT STACK
0A57                   *  RETURN RESULT IN PLACE OF X
0A57                   *
0A57 2A 1C 19          ASQR      LHLD     ASTKA
0A5A 11 C0 18                    LXI      D,FTEMP
0A5D CD 9C 0C                    CALL     VCOPY        SAVE X IN FTEMP
0A60                   * COMPUTE EXPONENT OF FIRST GUESS AS EXPONENT OF X/2
0A60 2A 1C 19                    LHLD     ASTKA
0A63 7E                          MOV      A,M
0A64 B7                          ORA      A
0A65 C8                          RZ       .            X=0
0A66 D6 80                       SUI      128
0A68 FA 71 0A                    JM       SQR5         NEGATIVE EXPONENT
0A6B 0F                          RRC
0A6C E6 7F                       ANI      127
0A6E C3 78 0A                    JMP      SQR6
0A71 2F                SQR5      CMA
0A72 3C                          INR      A
0A73 0F                          RRC
0A74 E6 7F                       ANI      127
0A76 2F                          CMA
0A77 3C                          INR      A
0A78 C6 80             SQR6      ADI      128
0A7A 77                          MOV      M,A
0A7B                   * TEST FOR NEGATIVE ARGUMENT
0A7B 2B                          DCX      H
0A7C 7E                          MOV      A,M
```

```
0A7D 01 41 4E          LXI    B,'NA'
0A80 B7                ORA    A
0A81 C2 C5 00          JNZ    ERROR      NEG ARGUMENT
0A84                 * DO NEWTON ITERATIONS
0A84                 * NEWGUESS =( X/OLDGUESS + OLDGUESS ) /2
0A84 3E 06             MVI    A,6        DO 6 ITERATIONS
0A86 F5        SQR20   PUSH   PSW        SET NEW ITERATION COUNT
0A87 01 C5 18          LXI    B,FTEM1
0A8A 11 C0 18          LXI    D,FTEMP    FTEMP IS 'X'
0A8D 2A 1C 19          LHLD   ASTKA      GUESS
0A90 CD AB 09          CALL   ADIV1      FTEM1=X/GUESS
0A93 11 C5 18          LXI    D,FTEM1
0A96 2A 1C 19          LHLD   ASTKA
0A99 44                MOV    B,H
0A9A 4D                MOV    C,L
0A9B CD 8D 09          CALL   AADD1      TOS=(X/GUESS)+GUESS
0A9E CD 2A 0B          CALL   PREPOP
0AA1 21 C5 07          LXI    H,FPTWO
0AA4 CD AB 09          CALL   ADIV1      TOS= (X/GUESS+GUESS)/2
0AA7 F1                POP    PSW
0AA8 3D                DCR    A          DECREMENT COUNT
0AA9 C2 86 0A          JNZ    SQR20      DO ANOTHER ITERATION
0AAC C9                RET
0AAD                 *
0AAD                 *      COMPUTE RND(X) X=TOP OF ARGUMENT STACK
0AAD                 * FRAND IS UPDATED TO NEW RANDOM VALUE
0AAD                 * A RANDOM NUMBER IN THE RANGE 0<RND<1 IS RETURNED IN PLACE
0AAD                 *
0AAD CD 2A 0B  ARND    CALL   PREPOP
0AB0 11 CF 18          LXI    D,FRAND
0AB3 21 CF 18          LXI    H,FRAND
0AB6 CD A1 09          CALL   AMUL1      TOS=FRAND*FRAND
0AB9                 * SET EXPONENT=0
0AB9 2A 1C 19          LHLD   ASTKA
0ABC 36 80             MVI    M,128      EXPONENT=128 (0 IN EXTERNAL FORM)
0ABE                 * PERMUTE DIGITS OF X AS
0ABE                 * 123456 INTO 345612
0ABE 01 FC FF          LXI    B,-4
0AC1 09                DAD    B
0AC2 46                MOV    B,M        SAVE 12
0AC3 23                INX    H
0AC4 23                INX    H
0AC5 CD FE 0A          CALL   PERMU      56=12
0AC8 CD FE 0A          CALL   PERMU      34=56
0ACB CD FE 0A          CALL   PERMU      12=34
0ACE                 * NORMALIZE NUMBER
0ACE 2A 1C 19  RND5    LHLD   ASTKA      TOS
0AD1 01 FC FF          LXI    B,-FPSIZ+1
0AD4 09                DAD    B
0AD5 7E                MOV    A,M        FIRST DIGIT PAIR
0AD6 E6 F0             ANI    15*16
0AD8 C2 F4 0A          JNZ    RND10      NUMBER IS NORMALIZED
0ADB                 * SHIFT LEFT ONE DIGIT
0ADB 2A 1C 19          LHLD   ASTKA
0ADE 7E                MOV    A,M        EXPONENT
0ADF 3D                DCR    A
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
0AE0 32 29 17            STA      EXP
0AE3 CD 84 16            CALL     LOAD       TOS INTO TEMP
0AE6 06 04               MVI      B,4
0AE8 CD CE 16            CALL     LEFT       SHIFT LEFT
0AEB CD 2A 0B            CALL     PREPOP
0AEE CD 9A 16            CALL     STORE
0AF1 C3 CE 0A            JMP      RND5       TEST IF NORMALIZED
0AF4                * SAVE NEW RANDM NUMBER IN FRAND CELL
0AF4 11 CF 18       RND10    LXI      D,FRAND
0AF7 2A 1C 19            LHLD     ASTKA
0AFA CD 9C 0C            CALL     VCOPY      FRAND=TOS
0AFD C9                  RET
0AFE                * PERMUTE PAIR OF DIGIT PAIRS
0AFE 7E             PERMU    MOV      A,M
0AFF 70                  MOV      M,B
0B00 47                  MOV      B,A
0B01 2B                  DCX      H
0B02 C9                  RET
0B03                *
0B03                *   EVALUATE P(X) USING HORNERS METHOD (X IS IN FTEMP)
0B03                * COEFFICIENT LIST POINTER IS IN BC
0B03                * RESULT REPLACES NUMBER ON TOP OF ARGUMENT STACK (Y)
0B03 2A 1C 19       POLY     LHLD     ASTKA
0B06 EB                  XCHG     .          DE=PTR TO Y
0B07 60                  MOV      H,B
0B08 69                  MOV      L,C        HL PTR TO COEFFICIENT LIST
0B09 CD 9C 0C            CALL     VCOPY      Y=FIRST COEFFICIENT
0B0C                * MULTIPLY BY X
0B0C E5             POLY1    PUSH     H          SAVE COEFF LIST POINTER
0B0D CD 2A 0B            CALL     PREPOP
0B10 21 C0 18            LXI      H,FTEMP
0B13 CD A1 09            CALL     AMUL1      Y=Y*X
0B16                * ADD NEXT COEFF
0B16 CD 2A 0B            CALL     PREPOP
0B19 E1                  POP      H
0B1A E5                  PUSH     H          HL=COEFF. LIST POINTER
0B1B CD 8D 09            CALL     AADD1      Y=Y+COEFF.
0B1E                * BUMP POINTER TO NEXT COEFFICIENT
0B1E E1                  POP      H          COEFF. POINTER
0B1F 01 FA FF            LXI      B,-FPSIZ-1
0B22 09                  DAD      B          NEXT COEF SIGN
0B23 7E                  MOV      A,M
0B24 23                  INX      H          PTR TO EXPONENT
0B25 B7                  ORA      A
0B26 F2 0C 0B            JP       POLY1      PROCESS NEXT COEFFICIENT
0B29 C9                  RET      .          NEGATIVE SIGN (-1) - ENDS LIST
0B2A                *
0B2A                * PREPARE FOR OPERATION
0B2A                *
0B2A 2A 1C 19       PREPOP   LHLD     ASTKA
0B2D EB                  XCHG     .          DE=ASTKA
0B2E 42                  MOV      B,D
0B2F 4B                  MOV      C,E
0B30 C9                  RET
0B31                *
0B31                *      QUADRANT COMPUTATION
```

```
0B31                * POPS TOP OF ARGUMENT STACK
0B31                * COMPUTE/GETS SIGN OF ARGUMENT, QUADRANT OF ARGUMENT
0B31                * AND INDEX INTO QUADRANT
0B31                *
0B31                *    EXITS WITH:
0B31                * SP POINTING TO QUADRANT, MOD 4
0B31                * SP+2 POINTING TO SIGN OF ARGUMENT
0B31                * TOP OF ARGUMENT STACK HAS INDEX INTO QUADRANT
0B31 2A 1C 19       QUADC    LHLD     ASTKA
0B34 2B                      DCX      H          POINT TO SIGN
0B35 46                      MOV      B,M
0B36 AF                      XRA      A
0B37 77                      MOV      M,A        ARG. SIGN=0
0B38 60                      MOV      H,B
0B39 E3                      XTHL     .          PUT SIGN ON STACK, POP RETURN
0B3A E5                      PUSH     H          PUSH RETURN
0B3B                * COMPUTE QUADRANT OF ABS(X)
0B3B 2A 1C 19                LHLD     ASTKA
0B3E CD A7 0C                CALL     PSHAS      PUT COPY OF ARG. ONTO STACK
0B41 CD 2A 0B                CALL     PREPOP
0B44 21 CF 07                LXI      H,PIC1     2/PI
0B47 CD A1 09                CALL     AMUL1      TOS=X*2/PI
0B4A CD 2A 0B                CALL     PREPOP
0B4D CD AA 0B                CALL     AINT       TOS=INT(X*2/PI)
0B50 2A 1C 19                LHLD     ASTKA
0B53 CD A7 0C                CALL     PSHAS      ANOTHER COPY
0B56 CD 8A 0D                CALL     PFIX       POP TOS TO DE
0B59 7B                      MOV      A,E
0B5A F5                      PUSH     PSW        QUADRANT
0B5B CD 2A 0B                CALL     PREPOP
0B5E 21 CA 07                LXI      H,PIC2
0B61 CD A1 09                CALL     AMUL1      TOS=INT(X*2/PI)
0B64 11 C0 18                LXI      D,FTEMP
0B67 CD BE 0C                CALL     POPA1      FTEMP=TOS
0B6A CD 2A 0B                CALL     PREPOP
0B6D 21 C0 18                LXI      H,FTEMP
0B70 CD 97 09                CALL     ASUB1      TOS=TOS-FTEMP
0B73 F1                      POP      PSW        A=QUADRANT, LOW ORDER BYTE
0B74 E6 03                   ANI      3          MOD 4
0B76 E1                      POP      H
0B77 F5                      PUSH     PSW        SAVE QUADRANT ON STACK
0B78 E9                      PCHL     .          RETURN
0B79                * SET UP ARG FOR USER CALL
0B79 CD 8A 0D       AARG     CALL     PFIX
0B7C EB                      XCHG
0B7D 22 4C 19                SHLD     CALLA
0B80 11 BB 18                LXI      D,FPSIN
0B83 C3 A8 0C                JMP      PSHA1      PUTS BACK THE ARG VALUE ON ARG STACK
0B86                * USED TO CALL USER ROUTINE
0B86 CD 8A 0D       ACALL    CALL     PFIX       GET THE ADDRESS
0B89 2A 4C 19                LHLD     CALLA      GET THE USER ARGUMENT
0B8C EB                      XCHG
0B8D 01 92 0B                LXI      B,ACAL1    RETURN LINK FOR USER ROUTINE
0B90 C5                      PUSH     B
0B91 E9                      PCHL
0B92 11 46 19       ACAL1    LXI      D,CALST
```

```
0B95 CD FA 0D                    CALL     CNS
0B98 3E 0D                       MVI      A,CR
0B9A 12                          STAX     D
0B9B 11 46 19                    LXI      D,CALST
0B9E 21 BB 18                    LXI      H,FPSIN
0BA1 CD 57 12                    CALL     FPIN
0BA4 11 BB 18                    LXI      D,FPSIN
0BA7 C3 A8 0C                    JMP      PSHA1       PUT THE RETURNED USER VALUE ON ARG STACK
0BAA                   *
0BAA                   *    INT FUNCTION ACTION ROUTINE
0BAA                   *
0BAA 0A                AINT      LDAX     B
0BAB D6 81                       SUI      129
0BAD F2 B3 0B                    JP       AINT1
0BB0                   * ZERO IF VALUE LESS THAN ONE
0BB0 AF                          XRA      A
0BB1 02                          STAX     B
0BB2 C9                          RET
0BB3                   * EXP > 0
0BB3 D6 05             AINT1     SUI      FPNIB-1
0BB5 D0                          RNC
0BB6 57                          MOV      D,A         COUNT
0BB7 0B                          DCX      B
0BB8 0B                AINT2     DCX      B
0BB9 0A                          LDAX     B
0BBA E6 F0                       ANI      360Q
0BBC 02                          STAX     B
0BBD 14                          INR      D
0BBE C8                          RZ
0BBF AF                          XRA      A
0BC0 02                          STAX     B
0BC1 14                          INR      D
0BC2 C2 B8 0B                    JNZ      AINT2
0BC5 C9                          RET
0BC6                   *
0BC6                   *      DIMENSION MATRIX
0BC6                   * SYMTAB ADDRESS IN HL, HL NOT CLOBBERED
0BC6                   * DE CONTAINS SIZE IN NUMBER OF ELEMENTS
0BC6                   *
0BC6 E5                DIMS      PUSH     H
0BC7 13                          INX      D
0BC8 D5                          PUSH     D
0BC9 21 00 00                    LXI      H,0
0BCC 0E 05                       MVI      C,FPSIZ
0BCE CD 18 0D                    CALL     RADD        MULTIPLY NELTS BY BYTES PER VALUE
0BD1 EB                          XCHG
0BD2 2A 1E 19                    LHLD     MATA
0BD5 E5                          PUSH     H
0BD6 19                          DAD      D
0BD7 CD E2 0C                    CALL     STOV        CHECK THAT STORAGE NOT EXHAUSTED
0BDA 22 1E 19                    SHLD     MATA        UPDATA MATRIX FREE POINTER
0BDD C1                          POP      B           BASE ADDR
0BDE D1                          POP      D           NELTS
0BDF E1                          POP      H           STMTAB ADDR
0BE0 E5                          PUSH     H
0BE1 72                          MOV      M,D
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
0BE2 2B                  DCX     H
0BE3 73                  MOV     M,E
0BE4 2B                  DCX     H
0BE5 70                  MOV     M,B
0BE6 2B                  DCX     H
0BE7 71                  MOV     M,C       SYMTAB ENTRY NOW SET UP
0BE8 E1                  POP     H
0BE9 C9                  RET
0BEA            *
0BEA            *   FIND VARIABLE OPTIONALLY SUBSCRIPTED IN TEXT
0BEA            * SETS CARRY IF NOT FOUND
0BEA            * RETURNS ADDRESS OF VARIABLE IN HL
0BEA            * UPDATES TXA IF FOUND
0BEA            *
0BEA CD 80 0C   VAR      CALL    ALPHA
0BED D8                  RC
0BEE CD 4D 0C            CALL    NAME2
0BF1 CD 02 0D            CALL    GC
0BF4 FE E0               CPI     LPARRW
0BF6 CA FE 0B            JZ      VAR1      TEST IF SUBSCRIPTED
0BF9            * MUST BE SCALAR VARIABLE
0BF9 CD 58 0C            CALL    STLK      RETURNS ENTRY ADDRESS IN HL
0BFC B7                  ORA     A         CLEAR CARRY
0BFD C9                  RET
0BFE            * MUST BE SUBSCRIPTED
0BFE CD 0A 0D   VAR1     CALL    GCI       GOBBLE LEFT PAREN
0C01 3E 80               MVI     A,200Q
0C03 B1                  ORA     C
0C04 4F                  MOV     C,A       SET TYPE TO MATRIX
0C05 CD 58 0C            CALL    STLK
0C08 E5                  PUSH    H         SYMBOL TABLE
0C09 11 0A 00            LXI     D,10      DEFAULT MATRIX SIZE
0C0C DC C6 0B            CC      DIMS      DEFAULT DIMENSION MATRIX
0C0F CD D7 05            CALL    EXPRB     EVALUATE SUBSCRIPT EXPRESSION
0C12 CD 8A 0D            CALL    PFIX      DE NOW HAS INTEGER
0C15 06 29               MVI     B,')'
0C17 CD FA 0C            CALL    EATC      GOBBLE RIGHT PAREN
0C1A E1                  POP     H
0C1B 2B                  DCX     H
0C1C CD 35 0D            CALL    DCMP      BOUNDS CHECK INDEX
0C1F D2 BC 00            JNC     E5
0C22 2B                  DCX     H
0C23 2B                  DCX     H
0C24 CD 40 0D            CALL    LHLI      GET BASE ADDR
0C27 0E 05               MVI     C,FPSIZ
0C29 13                  INX     D         BECAUSE BASE ADDR IS TO ELEMENT -1
0C2A CD 18 0D            CALL    RADD      ADD INDEX, CLEAR CARRY
0C2D C9                  RET
0C2E            *
0C2E            *   JUNK ON END OF STATEMENT, TEST IF AT END OF FILE
0C2E            * DOES NOT CLOBBER DE
0C2E            * EATS CHARACTER AND LINE COUNT AFTER CR
0C2E            * LEAVES NEW TXA IN HL
0C2E            * SETS CARRY IF END OF FILE
0C2E            *
0C2E CD 0A 0D   JOE      CALL    GCI
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
0C31 FE 3B        1830          CPI     ';'
0C33 C8           1831          RZ
0C34 FE 0D        1832          CPI     CR
0C36 C2 AA 00     1833          JNZ     E1
0C39 7E           1834          MOV     A,M
0C3A 3D           1835          DCR     A
0C3B CA 45 0C     1836          JZ      JOE2
0C3E 23           1837          INX     H
0C3F 23           1838          INX     H
0C40 23           1839          INX     H         SKIP OVER COUNT AND LINE NUMBER
0C41 22 95 17     1840 JOE1     SHLD    TXA
0C44 C9           1841          RET
0C45 37           1842 JOE2     STC
0C46 C3 41 0C     1843          JMP     JOE1
0C49              1844 *
0C49              1845 *     GET NAME FROM TEXT
0C49              1846 * SETS CARRY IF NAME NOT FOUND
0C49              1847 * IF SUCCEEDS RETURNS NAME IN BC, C=0 IF NO DIGIT IN NAME
0C49 CD 80 0C     1848 NAME     CALL    ALPHA
0C4C D8           1849          RC
0C4D 47           1850 NAME2    MOV     B,A
0C4E 0E 00        1851          MVI     C,0
0C50 CD 8D 0C     1852          CALL    DIG
0C53 3F           1853          CMC
0C54 D0           1854          RNC
0C55 4F           1855          MOV     C,A
0C56 B7           1856          ORA     A         CLEAR CARRY
0C57 C9           1857          RET
0C58              1858 *
0C58              1859 *     SYMBOL TABLE LOOKUP
0C58              1860 * BC CONTAIN NAME AND CLASS
0C58              1861 * IF NOT FOUND THEN CREATE ZERO'ED ENTRY AND SET CARRY
0C58              1862 * HL HAS ADDRESS ON RET
0C58              1863 *
0C58 2A 52 19     1864 STLK     LHLD    MEMTOP
0C5B 11 F9 FF     1865          LXI     D,-STESIZ SET UP BASE AND INCREMENT FOR SEARCH LOOP
0C5E 7E           1866 STLK0    MOV     A,M
0C5F B7           1867          ORA     A
0C60 CA 72 0C     1868          JZ      STLK2     TEST IF END OF TABLE
0C63 B8           1869          CMP     B
0C64 C2 6E 0C     1870          JNZ     STLK1     TEST IF ALPHA COMPARES
0C67 2B           1871          DCX     H
0C68 7E           1872          MOV     A,M       LOOK FOR DIGIT
0C69 B9           1873          CMP     C
0C6A 2B           1874          DCX     H
0C6B C8           1875          RZ      .         CARRY CLEAR OO RET
0C6C 23           1876          INX     H
0C6D 23           1877          INX     H
0C6E 19           1878 STLK1    DAD     D         DIDN'T COMPARE, DECREMENT POINTER
0C6F C3 5E 0C     1879          JMP     STLK0
0C72              1880 * ADD ENTRY TO SYMTAB
0C72 70           1881 STLK2    MOV     M,B
0C73 2B           1882          DCX     H
0C74 71           1883          MOV     M,C
0C75 23           1884          INX     H
0C76 EB           1885          XCHG
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
0C77 19             DAD     D
0C78 22 B3 18       SHLD    STA       STORE NEW END OF STMTAB POINTER
0C7B 1B             DCX     D
0C7C 1B             DCX     D
0C7D EB             XCHG
0C7E 37             STC
0C7F C9             RET
0C80          *
0C80          *  GOBBLES NEXT TEXT CHARACTER IF ALPHABETIC
0C80          * SETS CARRY IF NOT
0C80          * NEXT CHAR IN ACC ON FAILURE
0C80          *
0C80 CD 02 0D ALPHA  CALL   GC
0C83 FE 41           CPI    'A'
0C85 D8              RC
0C86 FE 5B           CPI    'Z'+1
0C88 3F              CMC
0C89 D8              RC
0C8A C3 97 0C        JMP    DIGT1
0C8D          * GOBBLES NEXT TEXT CHAR IF DIGIT
0C8D          * SETS CARRY IF NOT
0C8D          * NEXT CHAR IN ACC ON FAILURE
0C8D CD 02 0D DIG    CALL   GC
0C90 FE 30           CPI    '0'
0C92 D8              RC
0C93 FE 3A           CPI    '9'+1
0C95 3F              CMC
0C96 D8              RC
0C97 23       DIGT1  INX    H
0C98 22 95 17        SHLD   TXA
0C9B C9              RET
0C9C          *
0C9C          *  COPYS FPSIZ BYTES AT ADDR HL TO ADDR DE
0C9C          * ON EXIT HL POINTS TO ADR-1 OF LAST BYTE COPIED
0C9C          *
0C9C 0E 05    VCOPY  MVI    C,FPSIZ
0C9E 7E       VCOP1  MOV    A,M
0C9F 12              STAX   D
0CA0 2B              DCX    H
0CA1 1B              DCX    D
0CA2 0D              DCR    C
0CA3 C2 9E 0C        JNZ    VCOP1
0CA6 C9              RET
0CA7          *
0CA7          * PUSH VALUE ADDRESSED BY HL ONTO ARG STACK
0CA7          * SETS ARGF, CLEARS CARRY
0CA7          *
0CA7 EB       PSHAS  XCHG
0CA8 2A 1C 19 PSHA1  LHLD   ASTKA
0CAB 01 FB FF        LXI    B,-FPSIZ
0CAE 09              DAD    B
0CAF 22 1C 19        SHLD   ASTKA     DECREMENT ARG STACK POINTER
0CB2 EB              XCHG
0CB3 CD 9C 0C        CALL   VCOPY
0CB6 3E 01           MVI    A,1
0CB8 32 93 17        STA    ARGF      CLEAR ARGF
```

```
0CBB B7                      ORA     A          CLEAR CARRY
0CBC C9                      RET
0CBD                   *
0CBD                   *    POP ARG STACK
0CBD                   * HL CONTAINS ADDRESS TO PUT POPED VALUE AT
0CBD                   *
0CBD EB                POPAS    XCHG
0CBE 2A 1C 19          POPA1    LHLD    ASTKA
0CC1 E5                         PUSH    H
0CC2 01 05 00                   LXI     B,FPSIZ
0CC5 09                         DAD     B
0CC6 22 1C 19                   SHLD    ASTKA      INCREMENT STACK POINTER
0CC9 E1                         POP     H
0CCA C3 9C 0C                   JMP     VCOPY
0CCD                   *
0CCD                   *    PUSH FRAME ONTO CONTROL STACK
0CCD                   * TAKES MINUS AMOUNT TO SUB FROM CSTKA IN DE
0CCD                   * DOES OVERFLOW TEST AND RETURNS OLD CSTKA-1
0CCD                   *
0CCD 2A B5 18          PSHCS    LHLD    CSTKA
0CD0 E5                         PUSH    H
0CD1 19                         DAD     D
0CD2 22 B5 18                   SHLD    CSTKA
0CD5 EB                         XCHG
0CD6 21 D0 07                   LXI     H,LCSTKA   ADDR CONTAINS CSTKL
0CD9 CD 35 0D                   CALL    DCMP
0CDC DA B6 00                   JC      E4
0CDF E1                         POP     H
0CE0 2B                         DCX     H
0CE1 C9                         RET
0CE2                   *
0CE2                   *    STORAGE OVERFLOW TEST
0CE2                   * TEST THAT VALUE IN HL IS BETWEEN MATA AND STA
0CE2                   * DOES NOT CLOBBER HL
0CE2                   *
0CE2 EB                STOV     XCHG
0CE3 21 1E 19                   LXI     H,MATA
0CE6 CD 35 0D                   CALL    DCMP
0CE9 DA F4 0C                   JC      E8
0CEC 21 B3 18                   LXI     H,STA
0CEF CD 35 0D                   CALL    DCMP
0CF2 EB                         XCHG
0CF3 D8                         RC
0CF4 01 4F 53          E8       LXI     B,'SO'
0CF7 C3 C5 00                   JMP     ERROR
0CFA                   *
0CFA                   * INCREMENT TXA IF NEXT NON-BLANK CHAR IS EQUAL TO B
0CFA                   * ELSE SYNTAX ERROR
0CFA                   *
0CFA CD 0A 0D          EATC     CALL    GCI
0CFD B8                         CMP     B
0CFE C8                         RZ
0CFF C3 AA 00                   JMP     E1
0D02                   *
0D02                   *  GET NEXT NON-BLANK CHAR INTO ACC
0D02                   *  INCREMENT PAST BLANKS ONLY
```

```
0D02            *
0D02 CD 0A 0D   GC      CALL    GCI
0D05 2B                 DCX     H
0D06 22 95 17           SHLD    TXA
0D09 C9                 RET
0D0A            *
0D0A            *   GET NEXT NON-BLANK TEXT CHAR AND INCREMENT TXA
0D0A            * DOES NOT CLOBBER DE, BC
0D0A            * RETURN CHAR IN ACC
0D0A            *
0D0A 2A 95 17   GCI     LHLD    TXA
0D0D 7E         GCIO    MOV     A,M
0D0E 23                 INX     H
0D0F FE 20              CPI     ' '
0D11 CA 0D 0D           JZ      GCIO
0D14 22 95 17           SHLD    TXA
0D17 C9                 RET
0D18            *
0D18            *    REPEAT ADD
0D18            * ADDS DE TO HL C TIMES
0D18            *
0D18 19         RADD    DAD     D
0D19 0D                 DCR     C
0D1A C2 18 0D           JNZ     RADD
0D1D C9                 RET
0D1E            *
0D1E            *   PRINT MESSAGE ADDRESSED BY HL
0D1E            * ENDS WITH CHARACTER PROVIDED IN C
0D1E            * RETURN IN HL ADDRESS OF TERMINATOR
0D1E            *
0D1E 0E 0D      PRNTCR  MVI     C,CR
0D20 C3 25 0D           JMP     PRN1
0D23 0E 22      PRNT    MVI     C,'"'
0D25 7E         PRN1    MOV     A,M         GET NEXT CHAR
0D26 47                 MOV     B,A         FOR CHOUT
0D27 B9                 CMP     C           END OF MESSAGE TEST
0D28 C8                 RZ
0D29 FE 0D              CPI     CR
0D2B CA AA 00           JZ      E1          NEVER PRINT A CR IN THIS ROUTINE
0D2E CD 95 0E           CALL    CHOUT
0D31 23                 INX     H
0D32 C3 25 0D           JMP     PRN1
0D35            *
0D35            *   16 BIT UNSIGNED COMPARE
0D35            * COMPARE DE AGAINST VALUE ADDRESSED BY HL
0D35            * CLOBBERS A ONLY
0D35            *
0D35 7B         DCMP    MOV     A,E
0D36 96                 SUB     M
0D37 23                 INX     H
0D38 7A                 MOV     A,D
0D39 9E                 SBB     M
0D3A 2B                 DCX     H
0D3B C0                 RNZ
0D3C 7B                 MOV     A,E
0D3D 96                 SUB     M
```

```
0D3E B7                      ORA     A          CLEAR CARRY
0D3F C9                      RET
0D40                  *
0D40                  *    INDIRECT LOAD HL THRU HL
0D40                  *
0D40 F5               LHLI   PUSH    PSW
0D41 7E                      MOV     A,M
0D42 23                      INX     H
0D43 66                      MOV     H,M
0D44 6F                      MOV     L,A
0D45 F1                      POP     PSW
0D46 C9                      RET
0D47                  *
0D47                  *    GET FP CONSTANT FROM TEXT
0D47                  * PUSHES VALUE ON ARG STACK AND SETS ARGF FLAG
0D47                  * SETS CARRY IF NOT FOUND
0D47                  *
0D47 2A 95 17         CONST  LHLD    TXA        PREPARE CALL FPIN
0D4A EB                      XCHG
0D4B 21 BB 18                LXI     H,FPSINK
0D4E CD 57 12                CALL    FPIN
0D51 D8                      RC
0D52 1B                      DCX     D
0D53 EB                      XCHG
0D54 22 95 17                SHLD    TXA        NOW POINTS TO TERMINATOR
0D57 11 BB 18                LXI     D,FPSINK
0D5A CD A8 0C                CALL    PSHA1
0D5D AF                      XRA     A
0D5E 3C                      INR     A          SET A TO 1 AND CLEAR CARRY
0D5F 32 93 17                STA     ARGF
0D62 C9                      RET
0D63                  *
0D63                  *  DIRECT STATEMENT CHECKING ROUTINE
0D63                  *
0D63 3A 94 17         DIRT   LDA     DIRF
0D66 B7                      ORA     A
0D67 C8                      RZ
0D68 01 49 44                LXI     B,'DI'
0D6B C3 C5 00                JMP     ERROR
0D6E                  *
0D6E                  *    FIND TEXT LINE WITH LINE NUMBER GIVEN IN DE
0D6E                  * RETURNS TEXT ADDRESS  COUNT BYTE IN HL
0D6E                  *
0D6E 2A 50 19         FINDLN LHLD    BOFA
0D71 06 00                   MVI     B,0
0D73 4E               FIND1  MOV     C,M
0D74 79                      MOV     A,C
0D75 FE 01                   CPI     EOF
0D77 CA 84 0D                JZ      LERR
0D7A 23                      INX     H
0D7B CD 35 0D                CALL    DCMP
0D7E 2B                      DCX     H
0D7F C8                      RZ
0D80 09                      DAD     B
0D81 C3 73 0D                JMP     FIND1
0D84 01 4E 4C         LERR   LXI     B,'LN'
```

```
0D87 C3 C5 00              JMP     ERROR
0D8A                 *
0D8A                 *    FIX FLOATING TO POSITIVE INTEGER
0D8A                 * RETURN INTEGER VALUE IN DE
0D8A                 * FP VALUE FROM TOP OF ARG STACK, POP ARG STACK
0D8A                 *
0D8A 2A 1C 19        PFIX  LHLD    ASTKA
0D8D 44                    MOV     B,H
0D8E 4D                    MOV     C,L
0D8F E5                    PUSH    H
0D90 CD AA 0B              CALL    AINT
0D93 21 BB 18              LXI     H,FPSINK
0D96 CD BD 0C              CALL    POPAS
0D99 E1                    POP     H
0D9A 4E                    MOV     C,M          EXPONENT
0D9B 2B                    DCX     H
0D9C 7E                    MOV     A,M          SIGN
0D9D B7                    ORA     A
0D9E C2 BC 00              JNZ     E5           NEGATIVE NO GOOD
0DA1 11 FC FF              LXI     D,-FPSIZ+1
0DA4 19                    DAD     D
0DA5 11 00 00              LXI     D,0
0DA8 79                    MOV     A,C
0DA9 B7                    ORA     A
0DAA C8                    RZ
0DAB 0D                    DCR     C            SET UP FOR LOOP CLOSE TEST
0DAC 23              PFIX1 INX     H
0DAD 7E                    MOV     A,M
0DAE 0F                    RRC
0DAF 0F                    RRC
0DB0 0F                    RRC
0DB1 0F                    RRC
0DB2 CD C6 0D              CALL    MUL10
0DB5 DA BC 00              JC      E5
0DB8 0D                    DCR     C
0DB9 F0                    RP
0DBA 7E                    MOV     A,M
0DBB CD C6 0D              CALL    MUL10
0DBE DA BC 00              JC      E5
0DC1 0D                    DCR     C
0DC2 FA AC 0D              JM      PFIX1
0DC5 C9                    RET
0DC6                 *
0DC6                 * TAKE NEXT DIGIT IN A (MASK TO 17Q), ACCUMULATE TO DE
0DC6                 * PRESERVES ALL BUT A, DE
0DC6                 *
0DC6 E5              MUL10 PUSH    H
0DC7 33                    INX     SP
0DC8 33                    INX     SP
0DC9 62                    MOV     H,D          GET ORIGINAL VALUE TO HL
0DCA 6B                    MOV     L,E
0DCB 29                    DAD     H            DOUBLE IT
0DCC D8                    RC
0DCD 29                    DAD     H            AGAIN
0DCE D8                    RC
0DCF 19                    DAD     D            PLUS ORIGINAL MAKES 5 TIMES ORIG
```

```
0DD0 D8                    RC
0DD1 29                    DAD      H          TIMES TWO MAKES TEN
0DD2 D8                    RC
0DD3 EB                    XCHG
0DD4 3B                    DCX      SP
0DD5 3B                    DCX      SP
0DD6 E1                    POP      H
0DD7 E6 0F                 ANI      17Q
0DD9 83                    ADD      E
0DDA 5F                    MOV      E,A
0DDB 7A                    MOV      A,D
0DDC CE 00                 ACI      0          PROPOGATE THE CARRY
0DDE 57                    MOV      D,A
0DDF C9                    RET
0DE0              *
0DE0              *      GET INTEGER FROM TEXT
0DE0              * SET CARRY IF NOT FOUND
0DE0              * RETURN INTEGER VALUE IN HL
0DE0              * RETURN TERMINATOR IN ACC
0DE0              *
0DE0 CD 8D 0C     INTGER   CALL     DIG
0DE3 D8                    RC
0DE4 11 00 00              LXI      D,0
0DE7 C3 F1 0D              JMP      INTG2
0DEA CD 8D 0C     INTG1    CALL     DIG
0DED 62                    MOV      H,D
0DEE 6B                    MOV      L,E
0DEF 3F                    CMC
0DF0 D0                    RNC
0DF1 D6 30        INTG2    SUI      '0'
0DF3 CD C6 0D              CALL     MUL10
0DF6 D2 EA 0D              JNC      INTG1
0DF9 C9                    RET
0DFA              *
0DFA              *      CONVERT INTEGER TO STRING
0DFA              * DE CONTAINS ADDRESS OF STRING, RETURN UPDATED VALUE IN DE
0DFA              * HL CONTAINS VALUE TO CONVERT
0DFA              *
0DFA AF           CNS      XRA      A          SET FOR NO LEADING ZEROES
0DFB 01 F0 D8              LXI      B,-10000
0DFE CD 1F 0E              CALL     RSUB
0E01 01 18 FC              LXI      B,-1000
0E04 CD 1F 0E              CALL     RSUB
0E07 01 9C FF              LXI      B,-100
0E0A CD 1F 0E              CALL     RSUB
0E0D 01 F6 FF              LXI      B,-10
0E10 CD 1F 0E              CALL     RSUB
0E13 01 FF FF              LXI      B,-1
0E16 CD 1F 0E              CALL     RSUB
0E19 C0                    RNZ
0E1A 3E 30                 MVI      A,'0'
0E1C 12                    STAX     D
0E1D 13                    INX      D
0E1E C9                    RET
0E1F              *
0E1F              *    TAKE VALUE IN HL
```

```
0E1F                * SUB MINUS NUMBER IN BE THE MOST POSSIBLE TIMES
0E1F                * PUT VALUE ON STRING AT DE
0E1F                * IF A=0 THEN DONT PUT ZERO ON STRING
0E1F                * RETURN NON-ZERO IN A IF PUT ON STRING
0E1F                *
0E1F D5             RSUB     PUSH    D
0E20 16 FF                   MVI     D,-1
0E22 E5             RSUB1    PUSH    H
0E23 33                      INX     SP
0E24 33                      INX     SP
0E25 14                      INR     D
0E26 09                      DAD     B
0E27 DA 22 0E                JC      RSUB1
0E2A 3B                      DCX     SP
0E2B 3B                      DCX     SP
0E2C E1                      POP     H
0E2D 42                      MOV     B,D
0E2E D1                      POP     D
0E2F B0                      ORA     B          A GETS 0 IF A WAS 0 AND B IS 0
0E30 C8                      RZ
0E31 3E 30                   MVI     A,'0'
0E33 80                      ADD     B
0E34 12                      STAX    D
0E35 13                      INX     D
0E36 C9                      RET
0E37                *
0E37                *        INPUT CHARACTER FROM TERMINAL
0E37                *
0E37 CD 90 0E       INCHAR   CALL    STATUS
0E3A CA 37 0E                JZ      INCHAR     LOOP UNTIL CHAR RECEIVED
0E3D DB 01                   IN      1
0E3F E6 7F                   ANI     177Q
0E41 FE 03                   CPI     ESC
0E43 CA 62 00                JZ      CMND1
0E46 FE 0A                   CPI     LF         IGNORE LINE FEEDS
0E48 CA 37 0E                JZ      INCHAR
0E4B FE 00                   CPI     0          IGNORE NULLS
0E4D CA 37 0E                JZ      INCHAR
0E50 47                      MOV     B,A
0E51 C9                      RET
0E52                *
0E52 CD 1C 10       INLO     CALL    CRLF
0E55 21 D3 18       INLINE   LXI     H,IBUF
0E58 0E 49                   MVI     C,LINLEN
0E5A CD 37 0E       INL1     CALL    INCHAR
0E5D FE 5F                   CPI     5FH
0E5F CA 7E 0E                JZ      INL2       DELETE
0E62 77                      MOV     M,A
0E63 CD 95 0E                CALL    CHOUT      ECHO
0E66 78                      MOV     A,B
0E67 FE 40                   CPI     '@'        LINE DELETION
0E69 CA 52 0E                JZ      INLO
0E6C 06 0A                   MVI     B,LF       IN CASE WE ARE DONE
0E6E FE 0D                   CPI     CR
0E70 CA 95 0E                JZ      CHOUT      DO LF THEN RETURN
0E73 23                      INX     H
```

```
0E74 0D                  DCR     C
0E75 C2 5A 0E            JNZ     INL1
0E78 01 4C 4C            LXI     B,'LL'
0E7B C3 C5 00            JMP     ERROR
0E7E 79         INL2     MOV     A,C
0E7F 06 07               MVI     B,BELL
0E81 FE 49               CPI     LINLEN
0E83 CA 8A 0E            JZ      INL3
0E86 06 5F               MVI     B,5FH
0E88 2B                  DCX     H
0E89 0C                  INR     C
0E8A CD 95 0E   INL3     CALL    CHOUT
0E8D C3 5A 0E            JMP     INL1
0E90            *
0E90 DB 00      STATUS   IN      0
0E92 E6 40               ANI     40H
0E94 C9                  RET
0E95            *
0E95            *               OUTPUT ROUTINES
0E95            * SENSE SWITCH 1 (A8) CONTROLS VDM OR STANDARD TERMINAL
0E95            *
0E95 DB FF      CHOUT    IN      0FFH       GET SENSE SWITCH INPUT
0E97 1F                  RAR
0E98 DA E4 0F            JC      TEROT      SWITCH UP IS STANDARD TERMINAL
0E9B            *
0E9B            *
0E9B            *   OUTPUT DRIVER FOR PROCESSOR TECHNOLOGY VIDEO DISPLAY
0E9B            *   MODULE.  SEE VDM MANUAL FOR ADDITIONAL CODE COMMENTS
0E9B            *
0E9B 78         ALSOU    MOV     A,B
0E9C FE 7F               CPI     7FH
0E9E C8                  RZ
0E9F FE 07               CPI     BELL
0EA1 CA E4 0F            JZ      TEROT      "DING" GOES THE STANDARD TERMINAL
0EA4            *
0EA4            *   THIS ROUTINE ALLOWS AN OUTPUT RATE SELECTABLE FROM
0EA4            *   THE KEYBOARD BETWEEN THREE CHARACTERS PER SECOND
0EA4            *   TO APROX 2000 LINES PER MINUTE.
0EA4            *
0EA4 E5         TIMER    PUSH    H
0EA5 D5                  PUSH    D
0EA6 C5                  PUSH    B
0EA7 3A 45 19            LDA     SPEED      GET DELAY TIME
0EAA 67                  MOV     H,A
0EAB 2E 80               MVI     L,80H
0EAD CD C3 0E            CALL    NUMCK      SEE IF NEW SPEED IS WANTED
0EB0 AF                  XRA     A
0EB1 2B         TIME     DCX     H
0EB2 BC                  CMP     H          ZERO YET?
0EB3 C2 B1 0E            JNZ     TIME
0EB6 C1                  POP     B          GET BACK CHR
0EB7 C5                  PUSH    B
0EB8 78                  MOV     A,B
0EB9 CD 03 0F            CALL    SCOUT      OUTPUT CHR TO SCREEN
0EBC C1                  POP     B
0EBD D1                  POP     D
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
0EBE E1                    POP     H
0EBF 78                    MOV     A,B
0EC0 C3 EE 0F              JMP     CHCHK        NOW PROCESS THE REST
0EC3                 *
0EC3                 *  CHECK FOR TIMER CONTROL VALUE
0EC3                 *
0EC3 3A 3F 19        NUMCK LDA     BRKCHR
0EC6 47                    MOV     B,A
0EC7 B7                    ORA     A
0EC8 CC 90 0E              CZ      STATUS
0ECB CA D3 0E              JZ      NINP
0ECE DB 01                 IN      1
0ED0 E6 7F                 ANI     7FH
0ED2 47                    MOV     B,A
0ED3 78              NINP  MOV     A,B
0ED4 B7              NUMC2 ORA     A
0ED5 C8                    RZ      .            NO NEW INPUT VALUE
0ED6 FE 3A                 CPI     '9'+1
0ED8 D2 F2 0E              JNC     WAIT
0EDB FE 31                 CPI     '1'
0EDD DA F2 0E              JC      WAIT
0EE0 E6 0F                 ANI     0FH          REMOVE ASCII BIAS
0EE2 4F                    MOV     C,A          SAVE DELAY NUMBER
0EE3 AF                    XRA     A
0EE4 37                    STC     .            INITIALIZE DELAY BIT IN CARRY
0EE5 32 45 19        LESS  STA     SPEED
0EE8 17                    RAL
0EE9 0D                    DCR     C
0EEA C2 E5 0E              JNZ     LESS
0EED AF              CBRK  XRA     A
0EEE 32 3F 19        SBRK  STA     BRKCHR
0EF1 C9                    RET
0EF2                 *
0EF2 FE 20           WAIT  CPI     20H
0EF4 DA EE 0E              JC      SBRK
0EF7 C2 ED 0E              JNZ     CBRK
0EFA CD 90 0E        WAIT2 CALL    STATUS       WAIT FOR KEYBOARD INPUT
0EFD CA FA 0E              JZ      WAIT2
0F00 C3 EE 0E              JMP     SBRK
0F03                 *
0F03                 *   OUTPUT CHR IN REG A TO SCREEN
0F03                 *
0F03 4F              SCOUT MOV     C,A
0F04 21 41 19              LXI     H,CCP        GET CURRENT CURSOR POSITION
0F07 46                    MOV     B,M
0F08 FE 0D                 CPI     CR           CARRIAGE RETURN?
0F0A CA 2F 0F              JZ      SCOT2
0F0D FE 5F                 CPI     5FH          BACKSPACE?
0F0F CA 58 0F              JZ      BKSPA
0F12 FE 01                 CPI     'A'-40H      CURSOR ON-OFF
0F14 CA 8E 0F              JZ      CURTG
0F17 FE 1A                 CPI     'Z'-40H      CLEAR SCREEN
0F19 CA 68 0F              JZ      INITS
0F1C FE 20                 CPI     20H
0F1E D8                    RC      .            NO OTHER CONTROL CHRS
0F1F                 *
```

```
0F1F                 *       STORE CHR IN VDM MEMORY
0F1F                 *
0F1F 3A 40 19                LDA     CLN         CURRENT LINE NUMBER
0F22 CD C9 0F                CALL    CLNA        FIND ADDRESS
0F25 71                      MOV     M,C         PUT IN THE CHR
0F26 3A 41 19                LDA     CCP         ADVANCE THE CURSOR
0F29 3C                      INR     A
0F2A FE 40                   CPI     64          LAST CHR IN SCREEN LINE?
0F2C C2 39 0F                JNZ     SCOT1
0F2F                 *  ADVANCE CURSOR LINE
0F2F 3A 40 19        SCOT2   LDA     CLN
0F32 CD DC 0F                CALL    CCUR
0F35 CD A0 0F                CALL    SCRL
0F38 97                      SUB     A
0F39 32 41 19        SCOT1   STA     CCP
0F3C 47                      MOV     B,A
0F3D 3A 40 19                LDA     CLN
0F40 E6 0F           SCUR    ANI     0FH
0F42 32 40 19                STA     CLN
0F45 CD C9 0F                CALL    CLNA
0F48 78                      MOV     A,B
0F49 32 41 19                STA     CCP
0F4C 3A 42 19                LDA     CURF
0F4F B7                      ORA     A
0F50 7E                      MOV     A,M
0F51 CA E0 0F                JZ      CCUR2
0F54 F6 80                   ORI     80H
0F56 77                      MOV     M,A
0F57 C9                      RET
0F58                 *  BACKSPACE AND ERASE LAST CHR
0F58 3A 40 19        BKSPA   LDA     CLN
0F5B CD DC 0F                CALL    CCUR
0F5E 2B                      DCX     H
0F5F 36 20                   MVI     M,' '       BLANK IT
0F61 05                      DCR     B
0F62 3A 40 19                LDA     CLN
0F65 C3 40 0F                JMP     SCUR
0F68                 *
0F68                 *    CLEAR SCREEN AND INITIALIZE PARAMETERS
0F68                 *
0F68 21 00 CC        INITS   LXI     H,VDMBASE
0F6B 01 00 04                LXI     B,1024
0F6E 36 20           IL2     MVI     M,' '
0F70 23                      INX     H
0F71 0B                      DCX     B
0F72 78                      MOV     A,B
0F73 B1                      ORA     C
0F74 C2 6E 0F                JNZ     IL2         LOOP UNTIL ALL 1K CHARACTER PLACES HAVE A SPACE
0F77                 *  SCREEN IS CLEAR NOW SET PARAMETERS
0F77 32 43 19                STA     BOSL
0F7A 32 44 19                STA     BOTL
0F7D 32 90 17                STA     PHEAD
0F80 32 41 19                STA     CCP
0F83 32 42 19                STA     CURF
0F86 3E 0F                   MVI     A,15
0F88 32 40 19                STA     CLN
```

```
0F8B CD BB 0F              CALL    VDMOT
0F8E               * TOGGLE CURSOR (ON--OFF---ON.....)
0F8E 3A 42 19      CURTG   LDA     CURF
0F91 EE 01                 XRI     1
0F93 32 42 19              STA     CURF
0F96 3A 41 19              LDA     CCP
0F99 47                    MOV     B,A
0F9A 3A 40 19              LDA     CLN
0F9D C3 40 0F              JMP     SCUR
0FA0               *
0FA0 21 44 19      SCRL    LXI     H,BOTL
0FA3 E5                    PUSH    H
0FA4 7E                    MOV     A,M
0FA5 34                    INR     M
0FA6 96                    SUB     M
0FA7 01 00 00              LXI     B,0
0FAA CD C9 0F              CALL    CLNA
0FAD 01 40 20              LXI     B,2040H
0FB0 70            SCRL2   MOV     M,B
0FB1 2C                    INR     L
0FB2 0D                    DCR     C
0FB3 C2 B0 0F              JNZ     SCRL2
0FB6 E1                    POP     H
0FB7 7E                    MOV     A,M
0FB8 E6 0F                 ANI     0FH
0FBA 77                    MOV     M,A
0FBB               *
0FBB 3A 43 19      VDMOT   LDA     BOSL
0FBE 07                    RLC
0FBF 07                    RLC
0FC0 07                    RLC
0FC1 07                    RLC
0FC2 21 44 19              LXI     H,BOTL
0FC5 B6                    ORA     M
0FC6 D3 C8                 OUT     VDMDEV
0FC8 C9                    RET
0FC9               *
0FC9               *       CALCULATE SCREEN ADDRESS AND RETURN IN HL
0FC9               *
0FC9 6F            CLNA    MOV     L,A
0FCA 3A 44 19              LDA     BOTL
0FCD 85                    ADD     L
0FCE 0F                    RRC
0FCF 0F                    RRC
0FD0 6F                    MOV     L,A
0FD1 E6 03                 ANI     3           LOW MOD MATH
0FD3 C6 CC                 ADI     VDMPAGE
0FD5 67                    MOV     H,A
0FD6 7D                    MOV     A,L
0FD7 E6 C0                 ANI     0C0H
0FD9 80                    ADD     B
0FDA 6F                    MOV     L,A
0FDB C9                    RET
0FDC               *
0FDC CD C9 0F      CCUR    CALL    CLNA
0FDF 7E                    MOV     A,M
```

```
0FE0 E6 7F         CCUR2    ANI      7FH
0FE2 77                     MOV      M,A
0FE3 C9                     RET
0FE4               *
0FE4               *     STANDARD TERMINAL OUTPUT DRIVER
0FE4               *
0FE4 DB 00         TEROT    IN       0
0FE6 E6 80                  ANI      80H
0FE8 CA 95 0E               JZ       CHOUT
0FEB 78                     MOV      A,B
0FEC D3 01                  OUT      1           TYPE IT
0FEE FE 0D         CHCHK    CPI      CR
0FF0 C2 F7 0F               JNZ      CHLF        NOT CR IS IT LF?
0FF3 AF                     XRA      A
0FF4 C3 03 10               JMP      PSTOR       RETURN PHEAD TO ZERO
0FF7               *
0FF7 FE 0A         CHLF     CPI      LF
0FF9 CA 07 10               JZ       NULCH       IF LINEFEED PROCESS THE NULLS
0FFC FE 20                  CPI      40Q         NO PHEAD INC IF CONTROL CHAR
0FFE D8                     RC
0FFF 3A 90 17               LDA      PHEAD
1002 3C                     INR      A
1003 32 90 17      PSTOR    STA      PHEAD
1006 C9                     RET
1007               *
1007 3A 92 17      NULCH    LDA      NULLCT      OUTPUT NULL CHARS
100A B7                     ORA      A
100B C8                     RZ
100C C5                     PUSH     B
100D 4F                     MOV      C,A
100E 06 00                  MVI      B,NULL
1010 CD 95 0E      CH2      CALL     CHOUT       OUTPUT COUNT "C" NULLS
1013 0D                     DCR      C
1014 C2 10 10               JNZ      CH2
1017 C1                     POP      B
1018 C9                     RET
1019 CD 1C 10      CRLF2    CALL     CRLF
101C 06 0D         CRLF     MVI      B,CR
101E CD 95 0E               CALL     CHOUT
1021 06 0A                  MVI      B,LF
1023 C3 95 0E               JMP      CHOUT
1026               *
1026               *     CHECK IF PANIC CHARACTER HAS BEEN HIT
1026               *
1026 3A 3F 19      PCHECK   LDA      BRKCHR
1029 B7                     ORA      A
102A CC 90 0E               CZ       STATUS
102D C8                     RZ
102E DB 01                  IN       1           GET LAST CHR INPUT
1030 E6 7F                  ANI      7FH
1032 FE 03                  CPI      ESC
1034 CA 52 04               JZ       STOP1
1037 32 3F 19               STA      BRKCHR
103A C9                     RET
103B               *
103B               *     GET INTEGER FROM TERMINAL
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
103B                2555 * DE CONTAINS STRING TO PRINT FIRST
103B                2556 * HL HAS 1 LESS THAN ACCEPTABLE LOWER BOUND
103B                2557 * THIS ROUTINE GOES TO START IF BAD NUMBER
103B                2558 * INTEGER VALUE RETURNED IN HL
103B                2559 *
103B E5             2560 GINT     PUSH     H
103C EB             2561          XCHG
103D 3A 90 17       2562          LDA      PHEAD
1040 B7             2563          ORA      A
1041 C4 1C 10       2564          CNZ      CRLF
1044 CD 23 0D       2565          CALL     PRNT
1047 CD 55 0E       2566          CALL     INLINE
104A 21 D3 18       2567          LXI      H,IBUF
104D 22 95 17       2568          SHLD     TXA
1050 CD E0 0D       2569          CALL     INTGER
1053 DA 00 00       2570          JC       START
1056 FE 0D          2571          CPI      CR
1058 C2 00 00       2572          JNZ      START
105B D1             2573          POP      D
105C 22 D3 18       2574          SHLD     IBUF       USE IBUF AS A TEMP
105F 21 D3 18       2575          LXI      H,IBUF
1062 CD 35 0D       2576          CALL     DCMP
1065 D2 00 00       2577          JNC      START
1068 2A D3 18       2578          LHLD     IBUF       GET THE VALUE BACK TO HL
106B 7E             2579          MOV      A,M
106C 2F             2580          CMA
106D 77             2581          MOV      M,A        TRY TO STORE THERE
106E BE             2582          CMP      M
106F C2 00 00       2583          JNZ      START      BAD OR MISSING MEMORY
1072 C9             2584          RET
1073                2585 *
1073                2586 *      OUTPUT FP NUMBER ADDRESSED BY HL
1073                2587 *
1073 01 FC FF       2588 FPOUT    LXI      B,-DIGIT-1
1076 09             2589          DAD      B
1077 44             2590          MOV      B,H
1078 4D             2591          MOV      C,L
1079 21 2F 19       2592          LXI      H,ABUF     OUTPUT BUFFER
107C 3A 3A 19       2593          LDA      INFES      OUTPUT FORMAT
107F 32 39 19       2594          STA      FES        STORE IT
1082 1E 03          2595          MVI      E,DIGIT
1084 36 00          2596          MVI      M,0        CLEAR ROUND-OFF OVERFLOW BUFFER
1086 23             2597          INX      H          ABUF+1
1087                2598 *
1087 0A             2599 NXT      LDAX     B          GET DIGIT AND UNPACK
1088 57             2600          MOV      D,A
1089 1F             2601          RAR
108A 1F             2602          RAR
108B 1F             2603          RAR
108C 1F             2604          RAR
108D E6 0F          2605          ANI      17Q        REMOVE BOTTOM DIGIT
108F 77             2606          MOV      M,A        STORE TOP DIGIT IN OUTPUT BUFFER (ABUF)
1090 23             2607          INX      H
1091 7A             2608          MOV      A,D        NOW GET BOTTOM DIGIT
1092 E6 0F          2609          ANI      17Q
1094 77             2610          MOV      M,A        STORE IT
```

```
1095 23             2611          INX    H
1096 03             2612          INX    B
1097 1D             2613          DCR    E
1098 C2 87 10       2614          JNZ    NXT
109B 0A             2615          LDAX   B
109C 32 29 19       2616          STA    FSIGN      STORE SIGN OF NUMBER
109F AF             2617          XRA    A
10A0 77             2618          MOV    M,A        CLEAR ROUND-OFF BUFFER (ABUF+13) 12 DIGIT NO RND
10A1 21 37 19       2619          LXI    H,XSIGN    EXPONENT SIGN STORE
10A4 77             2620          MOV    M,A        CLEAR XSIGN
10A5                2621 *
10A5 03             2622 FIX      INX    B          GET EXPONENT
10A6 0A             2623          LDAX   B
10A7 B7             2624          ORA    A          EXPONENT ZERO?
10A8 CA B6 10       2625          JZ     ZRO
10AB D6 80          2626          SUI    128        REMOVE NORMALIZING BIAS
10AD C2 B1 10       2627          JNZ    FIX2
10B0 34             2628          INR    M          INCREMENT XSIGN TO NE ATIVE FLAG (1)LATFR ZERO
10B1 F2 B7 10       2629 FIX2     JP     CHK13
10B4 2F             2630          CMA    .          ITS A NEGATIVE EXPONENT
10B5 34             2631          INR    M          INCREMENT XSIGN TO NEGATIVE (1)
10B6 3C             2632 ZRO      INR    A
10B7 21 38 19       2633 CHK13    LXI    H,EXPO     EXPONENT TEMP STORE
10BA 77             2634          MOV    M,A
10BB 5F             2635          MOV    E,A
10BC FE 06          2636          CPI    DIGIT*2
10BE 21 39 19       2637          LXI    H,FES      FORMAT TEMP BYTE
10C1 DA C8 10       2638          JC     CHKXO
10C4 3E 01          2639 CHK40    MVI    A,1        FORCE EXPONENTIAL PRINTOUT
10C6 B6             2640          ORA    M SET      FORMAT FOR XOUT
10C7 77             2641          MOV    M,A
10C8                2642 *
10C8 7E             2643 CHKXO    MOV    A,M        CHECK IF EXPONENTIAL PRINTOUT
10C9 1F             2644          RAR
10CA D2 DB 10       2645          JNC    CHKX3
10CD E6 0F          2646          ANI    17Q
10CF FE 06          2647          CPI    DIGIT*2
10D1 DA D6 10       2648          JC     CHKX2
10D4 3E 05          2649          MVI    A,DIGIT*2-1  MAX DIGITS
10D6 57             2650 CHKX2    MOV    D,A
10D7 3C             2651          INR    A
10D8 C3 21 11       2652          JMP    ROUND
10DB                2653 *
10DB E6 0F          2654 CHKX3    ANI    17Q        ADD EXPONENT AND DECIMAL PLACES
10DD 57             2655          MOV    D,A
10DE 83             2656          ADD    E
10DF FE 07          2657          CPI    DIGIT*2+1
10E1 47             2658          MOV    B,A
10E2 DA EB 10       2659          JC     CHKXN
10E5 7E             2660          MOV    A,M
10E6 E6 40          2661          ANI    100Q
10E8 C2 C4 10       2662          JNZ    CHK40
10EB                2663 *
10EB 3A 37 19       2664 CHKXN    LDA    XSIGN      CHECK EXPONENT SIGN
10EE B7             2665          ORA    A
10EF C2 F6 10       2666          JNZ    XNEG       ITS NEGATIVE
```

```
10F2 78          2667          MOV   A,B
10F3 C3 21 11    2668          JMP   ROUND
10F6             2669 *
10F6 7A          2670 XNEG     MOV   A,D          SUBTRACT EXPONENT AND DECIMAL PLACE COUNT
10F7 93          2671          SUB   E
10F8 D2 12 11    2672          JNC   XN2
10FB 3A 3A 19    2673 XN1      LDA   INFES
10FE B7          2674          ORA   A
10FF F2 48 12    2675          JP    ZERO
1102 E6 0E       2676          ANI   16Q
1104 CA 48 12    2677          JZ    ZERO
1107 0F          2678          RRC
1108 5F          2679          MOV   E,A
1109 1D          2680          DCR   E
110A 0E 01       2681          MVI   C,1
110C 21 2E 19    2682          LXI   H,ABUF-1
110F C3 83 11    2683          JMP   NRND
1112 CA FB 10    2684 XN2      JZ    XN1
1115 C3 21 11    2685          JMP   ROUND
1118             2686 *
1118             2687 *
1118 06 1F       2688 CLEAN    MVI   B,37Q        CLEAR FLAGS
111A A0          2689          ANA   B
111B FE 07       2690          CPI   DIGIT*2+1
111D D8          2691          RC
111E 3E 07       2692          MVI   A,DIGIT*2+1  MAX DIGITS OUT
1120 C9          2693          RET
1121             2694 *
1121             2695 * This routine is used to round data to the
1121             2696 * specified decimal place
1121 CD 18 11    2697 ROUND    CALL  CLEAN
1124 4F          2698          MOV   C,A
1125 06 00       2699          MVI   B,0
1127 21 30 19    2700          LXI   H,ABUF+1
112A 09          2701          DAD   B            GET ROUND-OFF ADDRESS
112B 22 22 19    2702          SHLD  ADDT
112E 7E          2703          MOV   A,M
112F FE 05       2704          CPI   5            ROUND IF >=5
1131 DA 48 11    2705          JC    TRL2-1
1134             2706 *
1134 2B          2707 LESS1    DCX   H
1135 34          2708          INR   M            ROUND UP
1136 7E          2709          MOV   A,M
1137 B7          2710          ORA   A
1138 CA 49 11    2711          JZ    TRL2
113B FE 0A       2712          CPI   10           CHECK IF ROUNDED NUMBER >9
113D C2 45 11    2713          JNZ   TRAIL
1140 36 00       2714          MVI   M,0
1142 C3 34 11    2715          JMP   LESS1
1145             2716 *
1145             2717 * This routine is used to eliminate trailing zeros
1145 2A 22 19    2718 TRAIL    LHLD  ADDT
1148 2B          2719          DCX   H
1149 3A 39 19    2720 TRL2     LDA   FES          CHECK IF TRAILING ZEROS ARE WANTED
114C 17          2721          RAL
114D DA 5D 11    2722          JC    FPRNT        YES- GO PRINT DATA
```

```
1150 7E          TRL3     MOV   A,M
1151 B7                   ORA   A          IS IT A ZERO?
1152 C2 5D 11             JNZ   FPRNT      NO - GO PRINT
1155 2B                   DCX   H
1156 0D                   DCR   C          YES- FIX OUTPUT DIGIT COUNT
1157 FA 48 12             JM    ZERO
115A C3 50 11             JMP   TRL3
115D                 *
115D                 * Here starts the print format routines
115D 21 2F 19    FPRNT    LXI   H,ABUF
1160 7E                   MOV   A,M        CHECK IF ROUNDED UP TO 1
1161 B7                   ORA   A
1162 CA 83 11             JZ    NRND       JUMP IF NOT
1165 06 01                MVI   B,1
1167 3A 37 19             LDA   XSIGN      IS EXPONENT NEGATIVE?
116A B7                   ORA   A
116B CA 70 11             JZ    POSR
116E 06 FF                MVI   B,-1
1170                 *
1170 3A 38 19    POSR     LDA   EXPO       GET EXPONENT
1173 B7                   ORA   A
1174 C2 7C 11             JNZ   PO2        IS IT ZERO? (E+0)
1177 32 37 19             STA   XSIGN
117A 06 01                MVI   B,1
117C 80          PO2      ADD   B          FIX EXPONENT COUNT
117D 32 38 19             STA   EXPO
1180 1C                   INR   E
1181 0C                   INR   C
1182 2B                   DCX   H
1183                 *
1183 23          NRND     INX   H
1184 79                   MOV   A,C
1185 FE 07                CPI   DIGIT*2+1  CHECK FOR MAXIMUM DIGITS OUT
1187 C2 8B 11             JNZ   NRND1
118A 0D                   DCR   C
118B 3A 29 19    NRND1    LDA   FSIGN      CHECK IF NEGATIVE NUMBER
118E 1F                   RAR
118F D2 98 11             JNC   PRIN2      GO OUTPUT RADIX AND NUMBER
1192 CD 43 12             CALL  NEG        OUTPUT (-)
1195 C3 9B 11             JMP   PRI21
1198                 *
1198 CD 4D 12    PRIN2    CALL  SPACE      OUTPUT A SPACE
119B 3A 39 19    PRI21    LDA   FES        GET OUTPUT FORMAT
119E 1F                   RAR   .          CHECK IF EXPONENTIAL OUTPUT FORMAT
119F DA D6 11             JC    XPRIN
11A2 3A 37 19             LDA   XSIGN      GET EXPONENT SIGN
11A5 B7                   ORA   A          CHECK IF NEGATIVE EXPONENT
11A6 CA C8 11             JZ    POSIT
11A9 79                   MOV   A,C
11AA B7                   ORA   A
11AB C2 B2 11             JNZ   PRIN4      OUTPUT RADIX AND NUMBER
11AE CD 48 12             CALL  ZERO       NO DIGITS AFTER RADIX, OUPUT ZERO AND DONE
11B1 C9                   RET
11B2                 *
11B2 CD 52 12    PRIN4    CALL  RADIX      PRINT DECIMAL POINT
11B5 AF                   XRA   A
```

```
11B6 B3              ORA   E
11B7 CA C1 11        JZ    PRIN5     JUMP IF NO ZEROS TO PRINT
11BA CD 48 12        CALL  ZERO      FORCE PRINT A ZERO
11BD 1D              DCR   E
11BE C2 B5 11        JNZ   PRIN4+3
11C1            *
11C1 CD 39 12   PRIN5  CALL  NOUT    PRINT ASCII DIGIT
11C4 C2 C1 11        JNZ   PRIN5
11C7 C9              RET
11C8            *
11C8 CD 39 12   POSIT  CALL  NOUT
11CB 1D              DCR   E         BUMP EXPONENT COUNT
11CC C2 C8 11        JNZ   POSIT
11CF 79              MOV   A,C       CHECK IF MORE DIGITS TO OUPUT
11D0 B7              ORA   A
11D1 C8              RZ    .         NO, DONE
11D2 F8              RM
11D3 C3 B2 11        JMP   PRIN4     NOW PRINT DECIMAL POINT
11D6            *
11D6            * Get here for exponential output format
11D6 CD 39 12   XPRIN  CALL  NOUT
11D9 CA E5 11        JZ    NDEC      INTEGER?
11DC CD 52 12        CALL  RADIX     NO.....PRINT DECIMAL POINT
11DF CD 39 12   XPRI2  CALL  NOUT
11E2 C2 DF 11        JNZ   XPRI2
11E5            *
11E5 06 45      NDEC   MVI   B,'E'   OUTPUT 'E'
11E7 CD 95 0E        CALL  CHOUT
11EA 3A 37 19        LDA   XSIGN
11ED B7              ORA   A
11EE CA FB 11        JZ    XPRI3
11F1 CD 43 12        CALL  NEG       PRINT EXPONENT SIGN (-)
11F4 3A 38 19        LDA   EXPO
11F7 3C              INR   A
11F8 C3 04 12        JMP   XOUT2
11FB 06 2B      XPRI3  MVI   B,'+'   EXPONENT (+)
11FD CD 95 0E        CALL  CHOUT
1200            *
1200            * This routine is used to convert the exponent
1200            * from binary to ASCII and print the result
1200 3A 38 19   XOUT   LDA   EXPO
1203 3D              DCR   A
1204 0E 64      XOUT2  MVI   C,100
1206 16 00           MVI   D,0
1208 CD 2E 12        CALL  CONV
120B FE 30           CPI   '0'       SKIP LEADING ZEROS
120D CA 14 12        JZ    XO21
1210 14              INR   D
1211 CD 95 0E        CALL  CHOUT
1214 7B         XO21   MOV   A,E
1215 0E 0A           MVI   C,10
1217 CD 2E 12        CALL  CONV
121A FE 30           CPI   '0'
121C C2 23 12        JNZ   XO3
121F 15              DCR   D
1220 C2 26 12        JNZ   XO4
```

```
1223 CD 95 0E   XO3     CALL    CHOUT
1226 7B         XO4     MOV     A,E
1227 C6 30              ADI     '0'         ADD ASCII BIAS
1229 47                 MOV     B,A
122A CD 95 0E           CALL    CHOUT
122D C9                 RET
122E 06 2F      CONV    MVI     B,'0'-1
1230 04                 INR     B
1231 91                 SUB     C
1232 D2 30 12           JNC     CONV+2
1235 81                 ADD     C
1236 5F                 MOV     E,A
1237 78                 MOV     A,B
1238 C9                 RET
1239            *
1239            * This routine adds ASCII bias to a BCD digit
1239            * and calls the output routine
1239 7E         NOUT    MOV     A,M
123A C6 30              ADI     '0'
123C 47                 MOV     B,A
123D CD 95 0E           CALL    CHOUT
1240 23                 INX     H
1241 0D                 DCR     C           DECREMENT TOTAL DIGITS OUT COUNT
1242 C9                 RET
1243            *
1243            * Common symbol loading routines
1243 06 2D      NEG     MVI     B,'-'
1245 C3 95 0E           JMP     CHOUT
1248 06 30      ZERO    MVI     B,'0'
124A C3 95 0E           JMP     CHOUT
124D 06 20      SPACE   MVI     B,' '
124F C3 95 0E           JMP     CHOUT
1252 06 2E      RADIX   MVI     B,'.'
1254 C3 95 0E           JMP     CHOUT
1257            *CONVERTS FP STRING AT DE, UPDATE DE PAST TERMINATOR
1257            *PUTS TERMINATATOR IN B, PUTS FP NUMBER AT ADDRESS IN HL
1257            *SETS CARRY IF NOT FOUND
1257 E5         FPIN    PUSH    H
1258 D5                 PUSH    D
1259 EB                 XCHG
125A 2B                 DCX     H
125B 22 20 19           SHLD    ADDS
125E 21 26 19           LXI     H,OPST      CLEAR TEMPORARY STORAGE AREAS AND BC BUFFER
1261 0E 09              MVI     C,DIGIT+6
1263 CD 1B 13           CALL    CLEAR
1266            *
1266 11 00 00   SCANC   LXI     D,0
1269 21 2A 19           LXI     H,BC        BC=PACK BUFFER
126C 22 24 19   SCANO   SHLD    BCADD       PACK BUFFER POINTER
126F 21 6F 12   SCANP   LXI     H,SCANP
1272 E5                 PUSH    H           USED FOR RETURN FROM OTHER ROUTINES
1273 AF                 XRA     A
1274 32 37 19           STA     XSIGN       CLEAR EXPONENT SIGN BYTE
1277            *
1277 CD ED 12   SCANG   CALL    IBSCN
127A DA A4 12           JC      SCANX       FOUND A NUMBER, GO PACK IT
```

```
127D FE 2E          2891           CPI     '.'        RADIX?
127F CA 95 12       2892           JZ      SCAN5      PROCESS RADIX POINTERS
1282 FE 45          2893           CPI     'E'        EXPONENT?
1284 CA 23 13       2894           JZ      EXCON      FOUND 'E', GO PROCESS EXPONENT NUMBER
1287                2895 *NOT A CHARACTER LEGAL IN NUMBER
1287 47             2896           MOV     B,A        MOVE TERMINATOR TO B
1288 3A 26 19       2897           LDA     OPST       CHECK IF ANY DIGITS YET
128B E6 10          2898           ANI     20Q
128D C2 01 13       2899           JNZ     ENTR2
1290                2900 *GET HERE IF LEGAL FP NUMBER NOT FOUND
1290 E1             2901 FPIN1     POP     H          SCANP LINK
1291 D1             2902           POP     D          TEXT POINTER
1292 E1             2903           POP     H          FP # ADDR
1293 37             2904           STC
1294 C9             2905           RET
1295                2906 *FOUND DECIMAL POINT
1295 AF             2907 SCAN5     XRA     A          FOUND RADIX PROCESS RADIX POINTERS FOR EXP
1296 B2             2908           ORA     D          ANY DIGITS YET?
1297 C2 9F 12       2909           JNZ     SCAN6
129A C6 C0          2910           ADI     300Q       SET ECNT - STOP COUNTING DIGITS
129C B3             2911           ORA     E          NO INT DIGITS, BIT 7 IS COUNT/DON'T COUNT FLAG
129D 5F             2912           MOV     E,A        BIT 6 IS NEGATIVE EXPONENT FLAG
129E C9             2913           RET
129F 3E 80          2914 SCAN6     MVI     A,200Q     SET ECNT TO COUNT DIGITS
12A1 B3             2915           ORA     E
12A2 5F             2916           MOV     E,A
12A3 C9             2917           RET
12A4                2918 *
12A4 E6 0F          2919 SCANX     ANI     17Q        FOUND NUMBER-REMOVE ASCII BIAS
12A6 47             2920           MOV     B,A
12A7 21 26 19       2921           LXI     H,OPST     SET FIRST CHARACTER FLAG
12AA 3E 30          2922           MVI     A,60Q
12AC B6             2923           ORA     M
12AD 77             2924           MOV     M,A
12AE AF             2925           XRA     A
12AF B0             2926           ORA     B          IS CHARACTER ZERO?
12B0 C2 BC 12       2927           JNZ     PACK
12B3 B2             2928           ORA     D          LEADING ZERO? IE; ANY INT DIGITS?
12B4 C2 BC 12       2929           JNZ     PACK
12B7 B3             2930           ORA     E
12B8 5F             2931           MOV     E,A
12B9 C8             2932           RZ      .          IF COUNTING YET,
12BA 1C             2933           INR     E          ECNT+1-COUNT ZEROS FOR EXPONTENT COUNT
12BB C9             2934           RET
12BC                2935 *
12BC                2936 * This subroutine BCD packs digits into reg BC
12BC                2937 *
12BC 7B             2938 PACK      MOV     A,E
12BD 17             2939           RAL
12BE DA C2 12       2940           JC      PACK1
12C1 1C             2941           INR     E
12C2 7B             2942 PACK1     MOV     A,E
12C3 32 28 19       2943           STA     ECNT       DIGIT COUNT FOR EXPONENT COUNT
12C6 14             2944           INR     D          TOTAL DIGIT COUNT (D ALSO HAS TOP/BOTM FLAG BIT 7)
12C7 7A             2945           MOV     A,D
12C8 E6 7F          2946           ANI     177Q       REMOVE TOP/BOTTOM FLAG
```

```
12CA FE 07                 CPI     DIGIT*2+1   LIMIT INPUT DIGITS
12CC D0                    RNC
12CD AF                    XRA     A
12CE B2                    ORA     D
12CF FA DF 12              JM      BOTM
12D2                *
12D2 F6 80          TOP    ORI     200Q        SET MSB FOR TOP FLAG
12D4 57                    MOV     D,A
12D5 78                    MOV     A,B
12D6 2A 24 19              LHLD    BCADD       GET BC ADDRESS
12D9 07                    RLC
12DA 07                    RLC
12DB 07                    RLC
12DC 07                    RLC
12DD 77                    MOV     M,A         SAVE CHR IN BC
12DE C9                    RET
12DF                *
12DF E6 7F          BOTM   ANI     177Q        STRIP MSB (BOTTOM FLAG)
12E1 57                    MOV     D,A
12E2 78                    MOV     A,B
12E3 2A 24 19              LHLD    BCADD
12E6 B6                    ORA     M           OR IN TOP NUMBER
12E7 77                    MOV     M,A         PUT NUMBER BACK IN BC
12E8 23                    INX     H
12E9 C1                    POP     B
12EA C3 6C 12              JMP     SCANO
12ED 2A 20 19       IBSCN  LHLD    ADDS        INPUT BUFFER POINTER
12F0 23                    INX     H           GET NEXT BYTE
12F1 7E                    MOV     A,M
12F2 FE 20                 CPI     ' '
12F4 CA F0 12              JZ      IBSCN+3
12F7 22 20 19              SHLD    ADDS        NOTE: THIS ROUTINE FALLS THROUGH TO BELOW
12FA                *THIS ROUNTINE CHECKS FOR ASCII NUMBERS (0-9)
12FA FE 3A          NMCHK  CPI     '9'+1
12FC D0                    RNC
12FD FE 30                 CPI     '0'
12FF 3F                    CMC
1300 C9                    RET
1301                *
1301                * This routine is used to adjust a number in BC BUFFER
1301                * AND RETURNS VALUE
1301 11 00 00       ENTR2  LXI     D,0
1304 C5             ENT1   PUSH    B           TERMINATOR
1305 CD 7E 13              CALL    FIXE        NORMALIZE FLOATING POINT NUMBER
1308 C1                    POP     B           TERMINATOR
1309 D1                    POP     D           SCANP LINK
130A D1                    POP     D           OLD TEXT ADDR
130B D1                    POP     D           RETURN ADDR
130C 0E 05                 MVI     C,DIGIT+2
130E 21 2E 19              LXI     H,BC+DIGIT+1
1311 CD 9C 0C              CALL    VCOPY
1314 2A 20 19              LHLD    ADDS
1317 EB                    XCHG
1318 13                    INX     D
1319 B7                    ORA     A
131A C9                    RET
```

```
131B              * This routine is used to clear storage areas
131B              * The starting address is in H&L and the count
131B              * is in reg C
131B AF           CLEAR   XRA     A
131C 77                   MOV     M,A
131D 23                   INX     H
131E 0D                   DCR     C
131F C2 1C 13             JNZ     CLEAR+1
1322 C9                   RET
1323              *
1323              * This routine converts the ASCII exponent of
1323              * number in the input buffer to binary, and
1323              * normalizes exponent according to the input
1323              * format of the number
1323 CD ED 12     EXCON   CALL    IBSCN       GET CHARACTER
1326 DA 48 13             JC      EXC3
1329 FE E3                CPI     PLSRW       CHECK FOR UNARY SIGNS
132B CA 42 13             JZ      EXC4
132E FE 2B                CPI     '+'
1330 CA 42 13             JZ      EXC4
1333 FE 2D                CPI     '-'
1335 CA 3D 13             JZ      EXC2
1338 FE E5                CPI     MINRW
133A C2 7A 13             JNZ     FPERR       NO SIGN OR NUMBER?
133D 3E 01        EXC2    MVI     A,1
133F 32 37 19             STA     XSIGN       SAVE SIGN
1342 CD ED 12     EXC4    CALL    IBSCN
1345 D2 7A 13             JNC     FPERR       NO NUMBER?
1348 CD 4E 13     EXC3    CALL    ASCDC       CONVERT ASCII TO BINARY
134B C3 04 13             JMP     ENT1        NORMALIZE NUMBER AND RETURN
134E              *
134E              * This routine converts ASCII to binary
134E              * Three consecutive numbers <128 may be converted
134E EB           ASCDC   XCHG
134F 21 00 00             LXI     H,0
1352 1A           ASC1    LDAX    D           GET CHR FROM INPUT BUFFER- NO SPACES ALLOWED
1353 CD FA 12             CALL    NMCHK       CHECK IF NUMBER '
1356 D2 69 13             JNC     ASC2
1359 D6 30                SUI     '0'         REMOVE ASCII BIAS
135B 44                   MOV     B,H
135C 4D                   MOV     C,L
135D 29                   DAD     H
135E 29                   DAD     H
135F 09                   DAD     B
1360 29                   DAD     H
1361 4F                   MOV     C,A
1362 06 00                MVI     B,0
1364 09                   DAD     B
1365 13                   INX     D
1366 C3 52 13             JMP     ASC1
1369 EB           ASC2    XCHG
136A 47                   MOV     B,A         SAVE TERMINATOR
136B 22 20 19             SHLD    ADDS        SAVE IBUF ADDRESS
136E 7A                   MOV     A,D
136F B7                   ORA     A
1370 C2 7A 13             JNZ     FPERR       TOO BIG >255
```

```
1373 7B                    MOV   A,E
1374 17                    RAL
1375 DA 7A 13              JC    FPERR        TOO BIG >127
1378 1F                    RAR
1379 C9                    RET
137A C1          FPERR     POP   B            ASCDC RET LINK
137B C3 90 12              JMP   FPIN1
137E             *
137E             * This routine normalizes the input number
137E EB          FIXE      XCHG
137F 3A 2A 19              LDA   BC
1382 B7                    ORA   A            IS IT ZERO?
1383 CA 8B 13              JZ    ZZ2
1386 CD 8F 13              CALL  CHKPN        SET EXPONENT POSITIVE/NEGATIVE
1389 C6 80                 ADI   200Q         ADD EXPONENT BIAS
138B 32 2E 19    ZZ2       STA   BC+DIGIT+1    STORE NORMALIZED EXPONENT IN BC
138E C9                    RET
138F             *
138F 3A 28 19    CHKPN     LDA   ECNT         GET EXPONENT COUNT-SET IN 'SCAN' ROUTINE
1392 5F                    MOV   E,A
1393 E6 3F                 ANI   77Q          STRIP BITS 7&8
1395 47                    MOV   B,A
1396 3A 37 19              LDA   XSIGN
1399 B7                    ORA   A
139A CA B1 13              JZ    LPOS         EXPONENT IS POSITIVE
139D 24                    INR   H            SET SIGN IN H ** THIS SHOULD BE INR H NOT INX H
139E 3E 40                 MVI   A,100Q       L IS NEGATIVE
13A0 A3                    ANA   E            CHECK IF E IS NEGATIVE
13A1 CA AC 13              JZ    EPOS
13A4 7D                    MOV   A,L          BOTH E&L NEGATIVE
13A5 68                    MOV   L,B
13A6 CD BD 13              CALL  BPOS+1
13A9 2F                    CMA
13AA 3C                    INR   A
13AB C9                    RET   .            BACK TO FIXE
13AC             *
13AC 7D          EPOS      MOV   A,L          E&L NEGATIVE
13AD 2F                    CMA
13AE 3C                    INR   A
13AF 80                    ADD   B
13B0 C9                    RET   .            TO FIXE
13B1             *
13B1 3E 40       LPOS      MVI   A,100Q       EXPONENT POSITIVE
13B3 A3                    ANA   E            IS E NEGATIVE?
13B4 CA BC 13              JZ    BPOS
13B7 78                    MOV   A,B
13B8 45                    MOV   B,L
13B9 C3 AD 13              JMP   EPOS+1
13BC             *
13BC 78          BPOS      MOV   A,B          E&L POSITIVE
13BD 85                    ADD   L
13BE F0                    RP
13BF             *
13BF E1                    POP   H
13C0 C3 7A 13              JMP   FPERR
13C3 10                    DB    1*16
```

```
13C4 00 00              DW      0
13C6 01                 DB      1
13C7 81       FPNONE    DB      129
13C8          *
13C8          * THIS PROGRAM WRITTEN BY
13C8          *         MICROTEC
13C8          *         990 E. ARQUES
13C8          *         SUNNYVALE, CA.  94086
13C8          *
13C8          * THIS PROGRAM IS A FOUR FUNCTION FLOATING POINT BCD
13C8          * MATH PACKAGE.
13C8          * EACH FUNCTION MAY BE EXPRESSED AS: BC=DE # HL
13C8          * <BC> = ADDRESS OF RESULT
13C8          * <DE> = ADDRESS OF 1ST ARGUMENT
13C8          * <HL> = ADDRESS OF 2ND ARGUMENT
13C8          * # IS ONE OF THE FUNCTIONS: +,-,X,/.
13C8          * ALL ADDRESSES ON ENTRY, POINT TO THE EXPONENT PART OF
13C8          * THE FLOATING POINT NUMBER.
13C8          * EACH FLOATING POINT NUMBER CONSISTS OF (2*DIGIT) PACKED
13C8          * DECIMAL DIGITS, A SIGN AND A BIASED BINARY EXPONENT.  THE
13C8          * EXPONENT RANGE IS 10**-127 TO 10**127.
13C8          * THE NUMBER ZERO IS REPRESENTED BY THE EXPONENT 0.
13C8          * THE NUMBERS ARE STORED IN MEMORY AS (DIGIT) BYTES OF
13C8          * OF DECIMAL DIGITS
13C8          * STARTING AT THE LOW ORDER ADDRESS
13C8          * ALL NUMBER ARE ASSUMED TO BE NORMALIZED.  THAT IS EACH
13C8          * NUMBER CAN BE REPRESENTED AS   F**E.
13C8          * WHERE  .1<=F<1.0 AND F IS THE
13C8          * EXPONENT.
13C8          *
13C8          *     FLOATING POINT ADDITION
13C8          *
13C8 C5       FADD      PUSH    B
13C9 CD 42 16           CALL    EXPCK       FETCH ARGUMENTS
13CC 0E 00              MVI     C,0
13CE 1B       ADSUM     DCX     D
13CF EB                 XCHG
13D0 3A 28 17           LDA     SIGN
13D3 AE                 XRA     M           FORM SIGN OF RESULT
13D4 47                 MOV     B,A
13D5 EB                 XCHG
13D6 1A                 LDAX    D
13D7 1B                 DCX     D
13D8 A9                 XRA     C
13D9 32 28 17           STA     SIGN
13DC 21 2A 17           LXI     H,RCTRL     ROUNDING CONTROL FLAG
13DF 7E                 MOV     A,M
13E0 B7                 ORA     A
13E1 23                 INX     H
13E2 7E                 MOV     A,M         GET ROUNDING DIGIT
13E3 CA EA 13           JZ      ADS8
13E6 07                 RLC
13E7 07                 RLC
13E8 07                 RLC
13E9 07                 RLC
13EA C6 B0    ADS8      ADI     0B0H        FORCE CARRY IF DIGIT > 5
```

```
13EC 78               MOV   A,B
13ED 1F               RAR
13EE DA 2E 14         JC    ADS1       HAVE SUBTRACTION
13F1 17               RAL   .          RESTORE CARRY
13F2 CD 0D 14         CALL  ADD        PERFORM ADDITION
13F5 D2 04 14         JNC   ADS2
13F8 06 04            MVI   B,4
13FA CD A8 16         CALL  RIGHT
13FD 21 29 17         LXI   H,EXP
1400 34               INR   M          INCREMENT EXPONENT
1401 CA F4 16         JZ    OVER
1404 C1         ADS2  POP   B          GET RESULTS ADDRESS
1405 CD 9A 16         CALL  STORE      SAVE RESULTS
1408 C9               RET
1409 E1         ZEREX POP   H
140A C3 04 14         JMP   ADS2
140D 21 27 17   ADD   LXI   H,BUF+DIGIT-1
1410 06 03            MVI   B,DIGIT
1412 1A         ADD1  LDAX  D
1413 8E               ADC   M
1414 27               DAA
1415 77               MOV   M,A
1416 2B               DCX   H
1417 1B               DCX   D
1418 05               DCR   B
1419 C2 12 14         JNZ   ADD1
141C D0               RNC
141D 34               INR   M
141E C9               RET
141F            *
141F            *     FLOATING POINT SUBTRACTION
141F            *
141F C5         FSUB  PUSH  B
1420 CD 42 16         CALL  EXPCK      GET ARGUMENTS
1423 3A 28 17         LDA   SIGN
1426 EE 01            XRI   1          COMPLEMENT SIGN
1428 32 28 17         STA   SIGN
142B C3 CE 13         JMP   ADSUM
142E 17         ADS1  RAL   .          RESTORE CARRY
142F 3F               CMC   .          COMPLEMENT FOR ROUNDING
1430 CD 82 14         CALL  SUB        SUBTRACT ARGUMENTS
1433 21 28 17         LXI   H,SIGN
1436 DA 4D 14         JC    ADS4
1439 7E               MOV   A,M        GET SIGN
143A EE 01            XRI   1          COMPLEMENT
143C 77               MOV   M,A
143D 2B         ADS7  DCX   H
143E 06 03            MVI   B,DIGIT
1440 3E 9A      ADS3  MVI   A,9AH
1442 9E               SBB   M          COMPLEMENT RESULT
1443 C6 00            ADI   0
1445 27               DAA
1446 77               MOV   M,A
1447 2B               DCX   H
1448 05               DCR   B
1449 3F               CMC
```

```
144A C2 40 14     3227          JNZ     ADS3
144D 21 25 17     3228 ADS4     LXI     H,BUF
1450 01 03 00     3229          LXI     B,DIGIT
1453 7E           3230 ADS5     MOV     A,M
1454 B7           3231          ORA     A
1455 C2 66 14     3232          JNZ     ADS6
1458 23           3233          INX     H
1459 04           3234          INR     B
145A 04           3235          INR     B
145B 0D           3236          DCR     C
145C C2 53 14     3237          JNZ     ADS5
145F AF           3238          XRA     A
1460 32 29 17     3239          STA     EXP
1463 C3 04 14     3240          JMP     ADS2
1466 FE 10        3241 ADS6     CPI     10H
1468 D2 6C 14     3242          JNC     ADS9
146B 04           3243          INR     B
146C 21 29 17     3244 ADS9     LXI     H,EXP
146F 7E           3245          MOV     A,M
1470 90           3246          SUB     B
1471 CA FA 16     3247          JZ      UNDER
1474 DA FA 16     3248          JC      UNDER
1477 77           3249          MOV     M,A
1478 78           3250          MOV     A,B
1479 07           3251          RLC
147A 07           3252          RLC
147B 47           3253          MOV     B,A
147C CD CE 16     3254          CALL    LEFT
147F C3 04 14     3255          JMP     ADS2
1482 21 27 17     3256 SUB      LXI     H,BUF+DIGIT-1
1485 06 03        3257          MVI     B,DIGIT
1487 3E 99        3258 SUB1     MVI     A,99H
1489 CE 00        3259          ACI     0
148B 96           3260          SUB     M
148C EB           3261          XCHG
148D 86           3262          ADD     M
148E 27           3263          DAA
148F EB           3264          XCHG
1490 77           3265          MOV     M,A
1491 2B           3266          DCX     H
1492 1B           3267          DCX     D
1493 05           3268          DCR     B
1494 C2 87 14     3269          JNZ     SUB1
1497 C9           3270          RET
1498              3271 *
1498              3272 *    FLOATING POINT MULTIPLY
1498              3273 *
1498 C5           3274 FMUL     PUSH    B
1499 7E           3275          MOV     A,M
149A B7           3276          ORA     A           ARGUMENT = 0?
149B CA B2 14     3277          JZ      FMUL1+2
149E 1A           3278          LDAX    D
149F B7           3279          ORA     A           ARGUMENT = 0?
14A0 CA B2 14     3280          JZ      FMUL1+2
14A3 86           3281          ADD     M           FORM RESULT EXPONENT
14A4 DA AD 14     3282          JC      FMOVR
```

```
14A7 F2 FA 16             JP    UNDER
14AA C3 B0 14             JMP   FMUL1
14AD FA F4 16      FMOVR  JM    OVER
14B0 D6 80         FMUL1  SUI   128         REMOVE EXCESS BIAS
14B2 32 29 17             STA   EXP         SAVE EXPONENT
14B5 1B                   DCX   D
14B6 2B                   DCX   H
14B7 1A                   LDAX  D
14B8 AE                   XRA   M           FORM RESULT SIGH
14B9 2B                   DCX   H
14BA 1B                   DCX   D
14BB E5                   PUSH  H
14BC 21 28 17             LXI   H,SIGN      GET SIGN ADDRESS
14BF 77                   MOV   M,A         SAVE SIGN
14C0 2B                   DCX   H
14C1 AF                   XRA   A
14C2 06 05                MVI   B,DIGIT+2
14C4 77            FMUL2  MOV   M,A         ZERO WORKING BUFFER
14C5 2B                   DCX   H
14C6 05                   DCR   B
14C7 C2 C4 14             JNZ   FMUL2
14CA 3A 29 17             LDA   EXP
14CD B7                   ORA   A
14CE CA 09 14             JZ    ZEREX
14D1 0E 03                MVI   C,DIGIT
14D3 21 05 17             LXI   H,HOLD1+DIGIT
14D6               * GET MULTIPLIER INTO HOLDING REGISTER
14D6 1A            FMUL3  LDAX  D
14D7 77                   MOV   M,A         PUT IN REGISTER
14D8 2B                   DCX   H
14D9 1B                   DCX   D
14DA 0D                   DCR   C
14DB C2 D6 14             JNZ   FMUL3
14DE 71                   MOV   M,C
14DF 2B                   DCX   H
14E0 06 FA                MVI   B,250       SET LOOP COUNT
14E2 11 04 00      FMUL4  LXI   D,DIGIT+1
14E5 4B                   MOV   C,E
14E6 19                   DAD   D
14E7 EB                   XCHG
14E8 19                   DAD   D           H,L=NEXT HOLDING REGISTER
14E9 04                   INR   B
14EA F2 1E 15             JP    FMUL8       FINISHED
14ED 1A            FMUL5  LDAX  D           GET DIGITS
14EE 8F                   ADC   A           TIMES 2
14EF 27                   DAA
14F0 77                   MOV   M,A         PUT IN HOLDING REGISTER
14F1 1B                   DCX   D
14F2 2B                   DCX   H
14F3 0D                   DCR   C
14F4 C2 ED 14             JNZ   FMUL5
14F7 04                   INR   B           INCRREMENT LOOP COUNT
14F8 C2 E2 14             JNZ   FMUL4
14FB               * FORM 10X BY ADDING 8X AND 2X
14FB               * FIRST GET 8X
14FB 23                   INX   H
```

```
14FC 11 12 17          LXI   D,HOLD5     NEXT HOLDING REGISTER
14FF 0E 04             MVI   C,DIGIT+1
1501 41                MOV   B,C
1502 7E        FMUL6   MOV   A,M
1503 12                STAX  D
1504 23                INX   H
1505 13                INX   D
1506 0D                DCR   C
1507 C2 02 15          JNZ   FMUL6
150A 21 09 17          LXI   H,HOLD2+DIGIT   GET 2X
150D 1B                DCX   D
150E 1A        FMUL7   LDAX  D
150F 8E                ADC   M           FORM 10X
1510 27                DAA
1511 12                STAX  D
1512 1B                DCX   D
1513 2B                DCX   H
1514 05                DCR   B
1515 C2 0E 15          JNZ   FMUL7
1518 06 F9             MVI   B,249
151A EB                XCHG
151B C3 E2 14          JMP   FMUL4
151E EB        FMUL8   XCHG
151F 23                INX   H
1520 36 04             MVI   M,DIGIT+1   SET NEXT LOOP COUNT
1522           * PERFORM ACCUMULATION OF PRODUCT
1522 C1        FMUL9   POP   B           GET MULTIPLIER
1523 21 22 17          LXI   H,HOLD8+DIGIT+1
1526 35                DCR   M           DECREMNT LOOP COUNT
1527 CA 54 15          JZ    FMU14       FINISHED
152A 0A                LDAX  B
152B 0B                DCX   B
152C C5                PUSH  B
152D 2B                DCX   H
152E EB                XCHG
152F 87        FMU10   ADD   A           CHECK FOR BIT IN CARRY
1530 DA 3E 15          JC    FMU11       FOUND A BIT
1533 CA 4C 15          JZ    FMU12       ZERO - FINISHED THIS DIGIT
1536 21 FC FF          LXI   H,-DIGIT-1
1539 19                DAD   D           POINT TO NEXT HOLDING REGISTER
153A EB                XCHG
153B C3 2F 15          JMP   FMU10
153E 4F        FMU11   MOV   C,A
153F B7                ORA   A           CLEAR CARRY
1540 CD 0D 14          CALL  ADD         ACCUMULATE PRODUCT
1543 1A                LDAX  D
1544 86                ADD   M
1545 27                DAA
1546 77                MOV   M,A
1547 79                MOV   A,C
1548 1B                DCX   D
1549 C3 2F 15          JMP   FMU10
154C           * ROTATE RIGHT 1 BYTE
154C 06 08     FMU12   MVI   B,8
154E CD A8 16          CALL  RIGHT
1551 C3 22 15          JMP   FMUL9
```

```
1554 3A 25 17     FMU14    LDA     BUF
1557 E6 F0                 ANI     0F0H        CHECK IF NORMALIZED
1559 CA 65 15              JZ      FMU17
155C 7A                    MOV     A,D
155D E6 F0                 ANI     0F0H
155F 21 27 17              LXI     H,SIGN-1
1562 C3 76 15              JMP     FMU18
1565 06 04        FMU17    MVI     B,4
1567 21 29 17              LXI     H,EXP
156A 35                    DCR     M
156B CA FA 16              JZ      UNDER
156E CD CE 16              CALL    LEFT        NORMALIZE
1571 7A                    MOV     A,D         GET DIGIT SHIFTED OFF
1572              * PERFORM ROUNDING
1572 0F                    RRC
1573 0F                    RRC
1574 0F                    RRC
1575 0F                    RRC
1576 FE 50        FMU18    CPI     50H
1578 DA 9A 15              JC      FMU16
157B 3C                    INR     A
157C E6 0F                 ANI     0FH
157E 0E 03                 MVI     C,DIGIT
1580 8E           FMU15    ADC     M
1581 27                    DAA
1582 77                    MOV     M,A
1583 3E 00                 MVI     A,0
1585 2B                    DCX     H
1586 0D                    DCR     C
1587 C2 80 15              JNZ     FMU15
158A              * CHECK FOR ROUNDING OVERFLOW
158A D2 04 14              JNC     ADS2        NO OVERFLOW
158D 23                    INX     H
158E 36 10                 MVI     M,10H
1590 21 29 17              LXI     H,EXP
1593 34                    INR     M
1594 C2 04 14              JNZ     ADS2
1597 C3 F4 16              JMP     OVER
159A              * ROUNDING NOT NEEDED
159A E6 0F        FMU16    ANI     0FH
159C 86                    ADD     M
159D 77                    MOV     M,A
159E C3 04 14              JMP     ADS2
15A1              *
15A1              *    FLOATING POINT DIVISION
15A1              *
15A1 C5           FDIV     PUSH    B
15A2 7E                    MOV     A,M         FETCH DIVISOR EXP
15A3 B7                    ORA     A           DIVIDE BY 0?
15A4 CA F4 16              JZ      DIVZ
15A7 1A                    LDAX    D
15A8 B7                    ORA     A           DIVIDEND = 0?
15A9 CA FF 16              JZ      INSP
15AC 96                    SUB     M
15AD DA B6 15              JC      DIVUN
15B0 FA F4 16              JM      OVER
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
15B3 C3 B9 15                JMP     FDI1
15B6 F2 FA 16        DIVUN   JP      UNDER
15B9 C6 81           FDI1    ADI     129         FORM QUOTIENT EXP
15BB 32 06 17                STA     EXPD
15BE EB                      XCHG
15BF D5                      PUSH    D
15C0 CD 84 16                CALL    LOAD        FETCH DIVIDEND
15C3 D1                      POP     D
15C4 EB                      XCHG
15C5 3A 28 17                LDA     SIGN
15C8 2B                      DCX     H
15C9 AE                      XRA     M           FORM QUOTIENT SIGN
15CA 32 05 17                STA     SIGND
15CD EB                      XCHG
15CE 1B                      DCX     D
15CF 01 02 17                LXI     B,HOLD1
15D2 2E 06           DIV0    MVI     L,DIGIT+DIGIT
15D4 C5              DIV1    PUSH    B
15D5 E5                      PUSH    H
15D6 0E 00                   MVI     C,0         QUOTIENT DIGIT = 0
15D8 37              DIV3    STC     .           SET CARRY
15D9 21 27 17                LXI     H,BUF+DIGIT-1
15DC 06 03                   MVI     B,DIGIT
15DE 3E 99           DIV4    MVI     A,99H
15E0 CE 00                   ACI     0
15E2 EB                      XCHG
15E3 96                      SUB     M
15E4 EB                      XCHG
15E5 86                      ADD     M
15E6 27                      DAA
15E7 77                      MOV     M,A
15E8 2B                      DCX     H
15E9 1B                      DCX     D
15EA 05                      DCR     B
15EB C2 DE 15                JNZ     DIV4
15EE 7E                      MOV     A,M
15EF 3F                      CMC
15F0 DE 00                   SBI     0
15F2 77                      MOV     M,A
15F3 1F                      RAR
15F4 21 03 00                LXI     H,DIGIT
15F7 19                      DAD     D
15F8 EB                      XCHG
15F9 0C                      INR     C           INCREMENT QUOTIENT
15FA 17                      RAL
15FB D2 D8 15                JNC     DIV3
15FE B7                      ORA     A           CLEAR CARRY
15FF CD 0D 14                CALL    ADD         RESTORE DIVIDEND
1602 21 03 00                LXI     H,DIGIT
1605 19                      DAD     D
1606 EB                      XCHG
1607 C5                      PUSH    B
1608 06 04                   MVI     B,4
160A CD CE 16                CALL    LEFT        SHIFT DIVIDEND
160D C1                      POP     B
160E 0D                      DCR     C
```

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608


```
160F E1                    POP   H
1610 61                    MOV   H,C
1611 C1                    POP   B
1612 7D                    MOV   A,L
1613 C2 25 16              JNZ   DIV5
1616 FE 06                 CPI   DIGIT+DIGIT
1618 C2 25 16              JNZ   DIV5
161B 21 06 17              LXI   H,EXPD
161E 35                    DCR   M
161F CC FA 16              CZ    UNDER
1622 C3 D2 15              JMP   DIV0
1625 1F           DIV5     RAR
1626 7C                    MOV   A,H
1627 D2 35 16              JNC   DIV6
162A 0A                    LDAX  B
162B 07                    RLC
162C 07                    RLC
162D 07                    RLC
162E 07                    RLC
162F 84                    ADD   H
1630 02                    STAX  B          STORE QUOTIEN
1631 03                    INX   B
1632 C3 36 16              JMP   DIV7
1635 02           DIV6     STAX  B          STORE QUOTIENT
1636 2D           DIV7     DCR   L          DECREMENT DIGIT COUNT
1637 C2 D4 15              JNZ   DIV1
163A 21 06 17              LXI   H,EXPD
163D C1                    POP   B
163E CD 9D 16              CALL  STORO
1641 C9                    RET
1642              * FETCH AND ALIGN ARGUMENTS FOR
1642              * ADDITION AND SUBTRACTION
1642 1A           EXPCK    LDAX  D
1643 96                    SUB   M          DIFFERENCE OF EXPS
1644 0E 00                 MVI   C,0
1646 D2 4D 16              JNC   EXPC1
1649 0C                    INR   C
164A EB                    XCHG
164B 2F                    CMA
164C 3C                    INR   A
164D 47           EXPC1    MOV   B,A
164E 1A                    LDAX  D
164F 32 29 17              STA   EXP
1652 78                    MOV   A,B
1653 FE 06                 CPI   DIGIT+DIGIT
1655 DA 5A 16              JC    EXPC2
1658 3E 06                 MVI   A,DIGIT+DIGIT
165A 07           EXPC2    RLC
165B 07                    RLC
165C 47                    MOV   B,A
165D E6 04                 ANI   4
165F 32 2A 17              STA   RCTRL      SET ROUDING CONTROL
1662 C5                    PUSH  B
1663 D5                    PUSH  D
1664 CD 84 16              CALL  LOAD       LOAD SMALLER VALUE
1667 3E 28                 MVI   A,8*DIGIT+16
```

```
1669 90                    SUB   B
166A FE 28                 CPI   8*DIGIT+16
166C CA 7E 16              JZ    EXPC3
166F E6 F8                 ANI   0F8H
1671 1F                    RAR
1672 1F                    RAR
1673 1F                    RAR
1674 83                    ADD   E
1675 5F                    MOV   E,A
1676 7A                    MOV   A,D
1677 CE 00                 ACI   0
1679 57                    MOV   D,A
167A 1A                    LDAX  D          GET ROUDNING DIGIT
167B 32 2B 17              STA   RDIGI      SAVE
167E CD A8 16     EXPC3    CALL  RIGHT      ALIGN VALUES
1681 D1                    POP   D
1682 C1                    POP   B
1683 C9                    RET
1684              *LOAD ARGUMENT INTO BUFFER
1684 11 28 17     LOAD     LXI   D,SIGN
1687 0E 04                 MVI   C,DIGIT+1
1689 2B                    DCX   H
168A 7E           LOAD1    MOV   A,M
168B 12                    STAX  D
168C 2B                    DCX   H
168D 1B                    DCX   D
168E 0D                    DCR   C
168F C2 8A 16              JNZ   LOAD1
1692 AF                    XRA   A
1693 12                    STAX  D
1694 1B                    DCX   D
1695 12                    STAX  D
1696 32 2B 17              STA   RDIGI      ZERO ROUNDING DIGIT
1699 C9                    RET
169A              * STORE RESULTS IN MEMORY
169A 21 29 17     STORE    LXI   H,EXP
169D 1E 05        STORO    MVI   E,DIGIT+2
169F 7E           STOR1    MOV   A,M
16A0 02                    STAX  B
16A1 0B                    DCX   B
16A2 2B                    DCX   H
16A3 1D                    DCR   E
16A4 C2 9F 16              JNZ   STOR1
16A7 C9                    RET
16A8              * SHIFT RIGHT NUMBER OF DIGITS
16A8              * IN B/4
16A8 0E 04        RIGHT    MVI   C,DIGIT+1
16AA 21 24 17     RIGH1    LXI   H,BUF-1
16AD 78                    MOV   A,B
16AE D6 08                 SUI   8          CHECK IF BYTE CAN BE SHIFTED
16B0 D2 C1 16              JNC   RIGH3
16B3 05                    DCR   B
16B4 F8                    RM
16B5 B7                    ORA   A
16B6 7E           RIGH2    MOV   A,M
16B7 1F                    RAR
```

```
16B8 77                     MOV     M,A
16B9 23                     INX     H
16BA 0D                     DCR     C
16BB C2 B6 16               JNZ     RIGH2
16BE C3 A8 16               JMP     RIGHT
16C1           * SHIFT RIGHT ONE BYTE
16C1 47        RIGH3        MOV     B,A
16C2 AF                     XRA     A
16C3 56        RIGH4        MOV     D,M
16C4 77                     MOV     M,A
16C5 7A                     MOV     A,D
16C6 23                     INX     H
16C7 0D                     DCR     C
16C8 C2 C3 16               JNZ     RIGH4
16CB C3 A8 16               JMP     RIGHT
16CE           * SHIFT LEFT NUMBER OF DIGITS
16CE           * IN B/4
16CE 0E 04     LEFT         MVI     C,DIGIT+1
16D0 21 27 17               LXI     H,SIGN-1
16D3 78        LEF1         MOV     A,B
16D4 D6 08                  SUI     8
16D6 D2 E7 16               JNC     LEF3
16D9 05                     DCR     B
16DA F8                     RM
16DB B7                     ORA     A
16DC 7E        LEF2         MOV     A,M
16DD 17                     RAL
16DE 77                     MOV     M,A
16DF 2B                     DCX     H
16E0 0D                     DCR     C
16E1 C2 DC 16               JNZ     LEF2
16E4 C3 CE 16               JMP     LEFT
16E7           * SHIFT LEFT ONE BYTE
16E7 47        LEF3         MOV     B,A
16E8 AF                     XRA     A
16E9 56        LEF4         MOV     D,M
16EA 77                     MOV     M,A
16EB 7A                     MOV     A,D
16EC 2B                     DCX     H
16ED 0D                     DCR     C
16EE C2 E9 16               JNZ     LEF4
16F1 C3 CE 16               JMP     LEFT
16F4           * SET FLAGS FOR OVERFLOW,UNDERFLOW,
16F4           * AND DIVIDE BY ZERO
16F4 01 50 46  OVER         LXI     B,'FP'
16F7 C3 C5 00               JMP     ERROR
16FA 3E FF     UNDER        MVI     A,-1
16FC 32 23 17               STA     ERRI
16FF 33        INSP         INX     SP
1700 33                     INX     SP
1701 C9                     RET
1702           DIVZ         EQU     OVER
1702           *
1702           *       FLOATING POINT RAM
1702           *
1702           HOLD1        DS      DIGIT+1
```

```
1706   HOLD2   DS    DIGIT+1
170A   HOLD3   DS    DIGIT+1
170E   HOLD4   DS    DIGIT+1
1712   HOLD5   DS    DIGIT+1
1716   HOLD6   DS    DIGIT+1
171A   HOLD7   DS    DIGIT+1
171E   HOLD8   DS    DIGIT+1
1722           DS    1
1723   ERRI    DS    1           ERROR FLAG
1724           DS    1
1725   BUF     DS    DIGIT WORKING BUFFER
1728   SIGN    DS    1           SIGN BIT
1729   EXP     DS    1           EXPONENT
172A   RCTRL   DS    1           ROUNDING CONTROL FLAG 1=MSD
172B   RDIGI   DS    1           ROUNDING DIGIT
172C   SIGND   EQU   HOLD1+DIGIT
172C   EXPD    EQU   HOLD1+DIGIT+1
172C   *
172C   *     SYSTEM RAM
172C   *
172C   EROM    DS    0
172C           DS    100
1790   CMNDSP  DS    0
1790   PHEAD   DS    1
1791   RELTYP  DS    1
1792   NULLCT  DS    1
1793   ARGF    DS    1
1794   DIRF    DS    1
1795   TXA     DS    2
1797   CSTKSZ  EQU   100
1797   ASTKSZ  EQU   FPSIZ*LINLEN/2
1797   CSTKL   DS    CSTKSZ
17FB   ASTKL   DS    ASTKSZ
18B1   RTXA    DS    2
18B3   STA     DS    2
18B5   CSTKA   DS    2
18B7   SINK    DS    FPSIZ-1
18BB   FPSINK  DS    1
18BC           DS    FPSIZ-1
18C0   FTEMP   DS    1
18C1           DS    FPSIZ-1
18C5   FTEM1   DS    1
18C6           DS    FPSIZ-1
18CA   FTEM2   DS    1
18CB           DS    FPSIZ-1
18CF   FRAND   DS    1
18D0   IBCNT   DS    1
18D1   IBLN    DS    2
18D3   IBUF    DS    LINLEN
191C   ASTKA   DS    2
191E   MATA    DS    2
1920   ADDS    DS    2
1922   ADDT    DS    2
1924   BCADD   DS    2
1926   OPST    DS    1
1927   OPSTR   DS    1
```

```
1928      ECNT     DS     1
1929      FSIGN    DS     1
192A      BC       DS     DIGIT+2
192F      ABUF     DS     DIGIT*2+2
1937      XSIGN    DS     1
1938      EXPO     DS     1
1939      FES      DS     1
193A      INFES    DS     1
193B      MAXL     DS     2
193D      INSA     DS     2
193F      *
193F      *    VDM GLOBAL
193F      *
193F      BRKCHR   DS     1             BREAK CHR STORAGE
1940      CLN      DS     1             CURRENT SCREEN LINE NUMBER
1941      CCP      DS     1             CURRENT CURSOR POSITION
1942      CURF     DS     1             CURSOR DISPLAY SWITCH
1943      BOSL     DS     1             BEGINNING OF SCREEN LINE
1944      BOTL     DS     1             BOTTOM OF SCREEN LINE
1945      *
1945      VDMDEV   EQU    0C8H          VDM PORT ADDRESS FOR SCROLL CONTROL
1945      VDMBASE  EQU    0CC00H        VDM SCREEN MEMORY ADDRESS
1945      VDMPAGE  EQU    VDMBASE/256
1945      SPEED    DS     1
1946      *
1946      *    SPECIAL INTERFACE GLOBAL
1946      *
1946      CALST    DS     6
194C      CALLA    DS     2
194E      EOFA     DS     2             END OF FILE ADDRESS
1950      BOFA     DS     2             START OF FILE ADDRESS
1952      MEMTOP   DS     2             STORAGE FOR LAST ASSIGNED MEMORY LOCATION
1954      *
1954      *
```